



***PROTOTYPE URINE ANALYZER TELEMETRY MENGGUNAKAN
SENSOR WARNA UNTUK MENDETEKSI PENYAKIT DIABETES
PADA SESEORANG***

PROYEK AKHIR

**Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan Guna Memperoleh
Gelar Ahli Madya Teknik**



RAHMAT BUDIANTO

NIM. 14507134026

**PROGRAM STUDI TEKNIK ELEKTRONIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA DAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2018**

LEMBAR PERSETUJUAN
PROYEK AKHIR

PROTOTYPE URINE ANALYZER TELEMETRY MENGGUNAKAN
SENSOR WARNA UNTUK MENDETEKSI PENYAKIT DIABETES
PADA SESEORANG

Oleh:

RAHMAT BUDIANTO

14507134026

Telah diperiksa dan disetujui oleh pembimbing
Untuk diuji



Yogyakarta, 21 Maret 2018

Mengetahui,
Kaprodik Teknik Elektronika

Menyetujui,
Pembimbing Proyek Akhir

Dr. Sri Waluyanti, M.Pd.
NIP. 19581218 198603 2 001

Dr. Sri Waluyanti, M.Pd.
NIP. 19581218 198603 2 001

**LEMBAR PENGESAHAN
PROYEK AKHIR**

***Prototype Urine Analyzer Telemetry Menggunakan Sensor Warna
untuk Mendeteksi Penyakit Diabetes pada Seseorang***

Dipersiapkan dan Disusun Oleh:

RAHMAT BUDIANTO

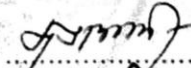


14507134026

Telah dipertahankan didepan Dewan Penguji Proyek Akhir
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Pada tanggal 22 Februari 2018

Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar
Ahli Madya Teknik

SUSUNAN DEWAN PENGUJI

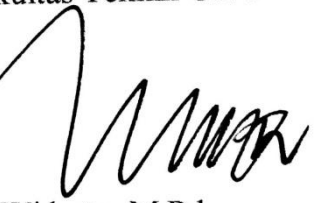
No.	Jabatan	Nama Lengkap	Tanda Tangan	Tanggal
1.	Ketua Penguji	Dr. Sri Waluyanti, M.Pd.		22/02 2018
2.	Sekretaris Penguji	Pipit Utami, M.Pd.		21/02 2018
3.	Penguji Utama	Dr. Fatchul Arifin, M.T.		22/02-2018

Yogyakarta, 21 Maret 2018

Mengetahui,



Dekan Fakultas Teknik UNY


Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Rahmat Budianto

NIM : 14507134026

Program Studi : Teknik Elektronika D-III


Judul Proyek Akhir : *Prototype Urine Analyzer Telemetry*

Menggunakan Sensor Warna untuk Mendeteksi
Penyakit Diabetes pada Seseorang

Dengan ini saya menyatakan proyek akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar ahli madya atau gelar lainnya di suatu Perguruan Tinggi, dan sepanjang pengetahuan penulis juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali secara tertulis diacu dalam masalah dan disebutkan dalam naskah dan disebutkan dalam daftar pustaka.

Yogyakarta, 21 Maret 2018

Yang menyatakan,



Rahmat Budianto

NIM. 14507134026

MOTTO

“Mensyukuri apa yang engkau miliki adalah syarat bagi kebahagiaanmu”

“Jangan takut untuk bermimpi. Karena mimpi adalah tempat menanam benih harapan dan memetakan cita-cita” Luffy

“Jalan seorang pria harus melewati darah dan air mata, karena penderitaanlah yang akan menempa hidup!!” Zoro

“Adalah tugas seorang pria untuk memaafkan kebohongan wanita” Sanji

“Hidup ini seperti pensil yang pasti akan habis, tetapi meninggalkan tulisan-tulisan yang indah dalam kehidupan” Nami

“Banyak orang ingin melupakan masa lalu, tapi sedikit orang yang belajar dari masa lalu” Nico Robin

“Kematian adalah hal yang menyakitkan tapi kematian itu pula yang dapat merubah orang lain” Chopper

“Tak ada gunanya memiliki sesuatu yang berharga jika kau tidak bisa melindunginya” Ace

LEMBAR PERSEMBAHAN

Proyek akhir ini saya persembahkan kepada :

Ayah, Ibu dan seluruh keluarga yang selalu mendampingi, memberi nasehat dan masukan, serta memfasilitasi saya hingga dapat mendorong dan memacu semangat saya.

Dosen Pembimbing proyek akhir, Dr.Sri Waluyanti, M.Pd yang selalu membimbing dalam penyelesaian proyek akhir ini.

Dosen Penasehat Akademik, Drs.Djoko Santoso, M.Pd yang selalu memotivasi untuk semangat dalam belajar di kampus UNY.

Sahabat kecilku Haun Istiqomah yang menjadi motivasi dan alasanku hingga sejauh ini.

Teman-teman kelas saya yaitu Teknik Elektronika 2014 UNY yang senantiasa membantu selama proses pengerjaan proyek akhir ini.

Sahabat dan teman-teman yang selalu menemani.

Kepada pihak-pihak yang telah membantu yang tidak bisa saya sebutkan satu persatu, terima kasih atas segala doa, bantuan dan inspirasi selama mengerjakan proyek akhir ini.

PROYEK AKHIR

Prototype Urine Analyzer Telemetry Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang

Oleh : Rahmat Budianto

NIM : 14507134026

ABSTRAK

Kurangnya kesadaran masyarakat untuk melakukan pemeriksaan kesehatan secara rutin setiap bulannya menyebabkan meningkatnya jumlah angka kematian setiap tahunnya. Padahal, pemeriksaan kesehatan perlu dilakukan mengingat saat ini banyak penyakit yang tidak selalu menimbulkan gejala awal. Sehingga hal ini mendorong masyarakat untuk terus merubah *mindset* dan giat melakukan pemeriksaan kesehatan secara rutin. Tujuan pembuatan alat ini yaitu untuk merancang dan mengetahui unjuk kerja dari rancangan yang telah dibuat.

Proses pembuatan alat terdiri dari beberapa tahapan yang memiliki fungsi masing-masing. Selain itu, pemilihan komponen yang digunakan juga telah disesuaikan dengan rancangan kebutuhan yang ada. Pada bagian *hardware*, Arduino UNO digunakan sebagai kontrol utama alat, Ethernet Shield sebagai perantara pengiriman data ke jaringan akan ditampilkan pada *webpage*, serta TCS230 sebagai sensor pendeteksi warna urine yang mana keseluruhan kerjanya tersebut disuplai dengan catu daya sebesar 8 VDC. Sedangkan pada bagian *software*, penggunaan Arduino IDE sebagai aplikasi pemrograman C/C++ dan Notepad++ sebagai aplikasi pemrograman HTML dipilih karena keduanya ini lebih mudah digunakan dan familiar oleh *user* dalam melakukan penulisan ataupun pengeditan program dengan jenis bahasanya masing-masing.

Berdasarkan hasil pengujian yang telah dilakukan dengan objek uji sejumlah 10 sampel, didapatkan hasil bahwa secara kualitas alat tersebut dapat diimplementasikan dan digunakan dengan baik. Sementara secara kuantitas diketahui bahwa rata-rata kesalahan pada setiap pendeteksian warna urine sebesar 10%. Sedangkan penampilan hasil pengukuran pada *website* saat pengujian tidak mengalami kendala. Kesalahan pengukuran umumnya disebabkan karena posisi sampel urine tidak sesuai, kurangnya database, serta tingkat sensitivitas dan akurasi dari komponen yang digunakan.

Kata kunci: Urine Analyzer, Sistem Telemetry, Arduino, TCS230.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat serta hidayat-Nya sehingga penulis dapat menyelesaikan laporan proyek akhir ini dengan judul “*Prototype Urine Analizer Telemetry Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang*”. Pembuatan proyek akhir sebagai syarat untuk memperoleh gelar Ahli Madya Fakultas Teknik Universitas Negeri Yogyakarta.

Penulis menyadari bahwa dalam menyelesaikan laporan ini tidak lepas dari bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ayah, Ibu dan seluruh keluarga yang telah memberikan fasilitas, dana dan dukungan kepada penulis.
2. Dr. Sri Waluyanti, M.Pd selaku Dosen Pembimbing dan Ketua Penguji proyek akhir sekaligus Ketua Program Studi Diploma III Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
3. Pipit Utami, S.Pd.T, M.Pd selaku Sekretaris Penguji proyek akhir.

4. Dr. Fatchul Arifin, M.T selaku Penguji Utama proyek akhir sekaligus Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
5. Drs. Djoko Santoso, M.Pd selaku Dosen Penasehat Akademik.
6. Dr. Widarto, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
7. Seluruh Dosen, Teknisi Lab dan Staff Jurusan Pendidikan Teknik Elektronika dan Informatika yang telah memberikan bantuan dan fasilitas sehingga terselesaikannya proyek akhir ini.
8. Teman-teman Teknik Elektronika angkatan 2014 Universitas Negeri Yogyakarta yang telah menyemangati selama pengerjaan proyek akhir.
9. Semua pihak yang telah membantu kami dalam pengerjaan laporan ini.

Demikian yang dapat saya sampaikan, kurang lebihnya saya mohon maaf. Dengan senang hati saya menerima kritik dan saran yang dapat membangun. Akhir kata penulis mengucapkan terima kasih dan semoga laporan ini dapat bermanfaat serta berguna bagi penulis khususnya, maupun pembaca pada umumnya.

Wassalamualaikum Wr. Wb.

Yogyakarta, 21 Maret 2018

Penulis,



Rahmat Budianto

NIM. 14507134026

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
 BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Identifikasi Masalah	3
C. Batasan Masalah	4
D. Rumusan Masalah	4
E. Tujuan	5
F. Manfaat	5
G. Keaslian Gagasan	6

BAB II PENDEKATAN PEMECAHAN MASALAH

A. Kadar Gula Darah pada Diabetes Melitus	9
B. <i>Urine Analysis</i>	14
C. TCS230	18
D. Arduino	20
E. Notepad++	32
F. Catu Daya	35

BAB III KONSEP RANCANGAN

A. Identifikasi Kebutuhan	38
B. Analisis Kebutuhan	38
C. Blok Diagram Rangkaian	40
D. Perancangan Sistem	41
E. Langkah Pembuatan Alat	45
F. Perangkat Lunak	49
G. Spesifikasi Alat	58
H. Pengujian Alat	58
I. Tabel Uji Alat	59
J. Pengoperasian Alat	61

BAB IV PENGUJIAN DAN PERSEMBAHAN

A. Pengujian	64
B. Pembahasan	73

BAB V KESIMPULAN DAN SARAN

A. Kesimpulan	81
---------------------	----

B. Keterbatasan Alat	82
C. Saran	83
DAFTAR PUSTAKA.....	84
LAMPIRAN.....	86

DAFTAR TABEL

	Halaman
Tabel 1. Kategori glukosa darah.....	10
Tabel 2. Jumlah urine berdasarkan usia.....	15
Tabel 3. Interpretasi warna urine	17
Tabel 4. Fungsi pin sensor warna TCS230	19
Tabel 5. Mode pemilihan photodiode pembacaan warna	20
Tabel 6. Spesifikasi Arduino Uno	23
Tabel 7. Tipe data bahasa C	29
Tabel 8. Operasi kondisi	30
Tabel 9. Karakteristik regulator tegangan seri 78XX	37
Tabel 10. Penentuan kondisi TCS230	55
Tabel 11. Hasil pengukuran <i>power supply</i> tegangan tanpa beban	59
Tabel 12. Hasil pengukuran <i>power supply</i> tegangan dengan beban	59
Tabel 13. Hasil pembacaan sensor TCS230 dalam model RGB	60
Tabel 14. Hasil pengujian tampilan pada <i>web browser</i>	60
Tabel 15. Hasil pengujian unjuk kerja pembacaan sampel urine	60
Tabel 16. Hasil pengukuran <i>power supply</i> tegangan tanpa beban	65
Tabel 17. Hasil pengukuran <i>power supply</i> tegangan dengan beban	66
Tabel 18. Hasil pembacaan sensor TCS230 dalam model RGB	67
Tabel 19. Hasil pengujian tampilan pada <i>web browser</i>	69
Tabel 20. Hasil pengujian unjuk kerja pembacaan sampel urine	70

DAFTAR GAMBAR

	Halaman
Gambar 1. Bagan warna Dipstick Urinalysis	16
Gambar 2. Contoh warna urine	16
Gambar 3. Bentuk fisik TCS230	19
Gambar 4. Konstruksi TCS230	19
Gambar 5. Arduino Uno R3	23
Gambar 6. Bentuk fisik Arduino Ethernet Shield	26
Gambar 7. Tampilan Arduino IDE	28
Gambar 8. Rangkaian dan IC regulator	37
Gambar 9. Blok diagram rangkaian.....	40
Gambar 10. Rangkaian sensor TCS230 yang terhubung ke Arduino	42
Gambar 11. Rangkaian <i>power supply</i> regulator 7808	44
Gambar 12. Tampilan pada com serial Arduino IDE.....	50
Gambar 13. Tampilan <i>web</i> pada PC <i>client</i>	51
Gambar 14. <i>Flowchart</i> program utama.....	52
Gambar 15. Program <i>filter</i> sensor warna pada Arduino IDE	54
Gambar 16. <i>Flowchart</i> program subrutin baca data sensor	56
Gambar 17. Program kirim data ke <i>client</i> pada Arduino IDE	57
Gambar 18. <i>Flowchart</i> program subrutin kirim data sensor ke <i>client</i>	57
Gambar 19. Jaringan <i>localhost</i> antara <i>prototype urine analyzer</i> <i>telemetry</i> dengan satu PC <i>client</i>	61
Gmabar 20. <i>Setting IP static</i> pada PC <i>client</i>	62

Gambar 21. <i>Interfaceweb</i> dari <i>browser PC client</i>	63
Gambar 22. Rangkaian <i>power supply</i> tanpa beban pada simulasi	
Proteus	65
Gambar 23. Rangkaian <i>power supply</i> dengan beban pada simulasi	
Proteus	65
Gambar 24. Sampel warna pengujian sensor	68

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Barang Habis Pakai	88
Lampiran 2. Data Kalibrasi Pembacaan Sensor TCS230 Sebelum Remaping ke Model RGB 255.....	89
Lampiran 3. Skema Rangkaian dan Layout PCB <i>Power Supply</i>	90
Lampiran 4. Skema Rangkaian dan Layout PCB <i>Prototype Urine Analyzer Telemetry</i>	91
Lampiran 5. List Program Arduino	92
Lampiran 6. List Program Halaman <i>Web</i>	98
Lampiran 7. Pengoprasian Alat dengan Switch dan Router	115
Lampiran 8. Data Sheet IC 7808	145
Lampiran 9. Data Sheet TCS230.....	155
Lampiran 10. Data Sheet Arduino UNO.....	167
Lampiran 11. Data Sheet Arduino Ethernet Shield.....	171
Lampiran 12. Data Pembacaan Sensor	176
Lampiran 13. Desain Alat	177
Lampiran 14. Foto Alat.....	178

BAB I

PENDAHULUAN

A. Latar Belakang

Penyakit Diabetes Melitus (DM) merupakan salah satu dari empat penyakit tidak menular tertinggi yang berakibat pada kematian, dengan tiga penyakit lainnya yaitu penyakit jantung, kanker, dan pembuluh darah. Selain itu penyakit diabetes juga dapat menyebabkan beberapa penyakit turunan seperti hipertensi dan penyakit jantung (Abidin, 2015). Saat ini diperkirakan sekitar 285 juta orang menderita diabetes di seluruh dunia dan diperkirakan akan meningkat menjadi 366 juta pada tahun 2030 (Kassim, 2013). Indonesia sendiri, pada tahun 2012 tercatat lebih dari 8 juta jiwa mengidap penyakit diabetes, dan 84% tidak menyadari bahwa mereka menderita diabetes (WHO, 2011).

Penyakit diabetes dapat berpengaruh terhadap otak dengan penderita beresiko meningkatnya penyakit Alzheimer dan perlambatan fungsi mental (Anna, 2011). Banyaknya bahaya yang dapat ditimbulkan dari penyakit diabetes melitus, menjadikan penyakit ini harus segera ditangani. Sejatinya tujuan pengobatan pada penyakit ini memang bukan untuk menyembuhkan, melainkan untuk menormalkan kadar gula darah dan mencegah kerusakan organ-organ tubuh akibat gula darah yang tinggi. Oleh sebab itu pengukuran kadar gula darah secara rutin perlu dilakukan agar seseorang dapat mengurangi resiko terhadap bahaya penyakit ini.

Saat ini terdapat beberapa cara yang dapat digunakan untuk mengukur kadar gula darah. Teknik pengukuran tersebut masing-masing memiliki kelebihan dan kekurangan baik itu dari segi akurasi, kompleksitas, maupun biaya. Secara garis besar teknik pengukuran gula darah terbagi menjadi dua metode, yaitu metode *invasive* dan *non invasive*. Metode *invasive* merupakan metode yang banyak digunakan untuk mengetahui kadar gula dalam darah. Namun karena pada metode *invasive* menggunakan sampel berupa darah pasien, sehingga metode ini kurang efektif dan cocok digunakan sebab terkadang dapat menimbulkan adanya infeksi dan rasa sakit yang dirasakan pasien setelahnya. Sedangkan metode *non invasive*, teknik ini menggunakan alat photoplethysmography. Untuk mengeceknya, sensor pendeteksi diletakkan di organ tubuh pasien seperti pada jari atau daun telinga, sehingga pengukuran ini dianggap kurang akurat.

Sementara pada pengukuran menggunakan sampel urine, hasil pengukurannya akan lebih akurat karena sejatinya urine mengandung zat-zat buangan tubuh yang dapat diketahui dengan jelas apakah seseorang menderita penyakit khususnya diabetes melitus ataupun tidak. Pendeteksian menggunakan urine dapat dilakukan dengan menganalisa kandungan kimia yang terdapat pada urine maupun dari warna urine yang tidak wajar. Namun, pengujian ini hanya dapat dilakukan di laboratorium dan dibutuhkan biaya yang tidak sedikit serta hanya dapat dilakukan oleh

tenaga ahli. Oleh sebab itu, diperlukan sistem yang terintegrasi untuk dapat menghasilkan hasil diagnosa yang cepat dan akurat.

Berdasarkan hal-hal tersebut diatas penulis mencoba membuat suatu alat *urine analyzer* dengan sistem telemetri. Rancangan ini diharapkan mampu mendeteksi penyakit diabetes melitus. Pengukurannya menggunakan indikator warna urine pada volume yang tetap.

B. Identifikasi Masalah

Dari uraian latar belakang di atas, maka dapat dibuat suatu identifikasi masalah sebagai berikut:

1. Penyakit diabetes melitus menjadi penyumbang tertinggi penyakit tidak menular di seluruh dunia.
2. Perlunya penanganan khusus terhadap penderita diabetes melitus sebab dapat terjadi gangguan yang lebih fatal apabila tidak segera diobati.
3. Cukup mahalnya biaya pengecekan penyakit diabetes menyebabkan sebagian masyarakat sulit untuk melakukannya.
4. Hasil diagnosa penyakit masih bersifat lama karena dilakukan secara manual.
5. Belum terciptanya alat *urine analyzer* yang terintegrasi dengan sistem telemetri untuk mendeteksi penyakit diabetes melitus yang terpantau menggunakan web server.

C. Batasan Masalah

Berdasarkan identifikasi masalah yang telah diuraikan di atas, maka perlu adanya batasan masalah sehingga ruang lingkup masalah menjadi jelas. Pada proyek akhir ini dibatasi pada poin5, yaitu belum terciptanya alat *urine analyzer* yang terintegrasi dengan sistem telemetri untuk mendeteksi penyakit diabetes melitus dengan cepat dan akurat. Cara kerja alat ini dengan menggunakan urine sebagai media *input* sensor. Sensor warna TCS230 digunakan untuk mendeteksi penyakit diabetes mellitus melalui perbedaan warna urine yang menjadi parameternya. Keseluruhan alat diolah dan dikendalikan oleh *chip* Arduino UNO dengan hasil deteksinya ditampilkan dalam tampilan *webpage local* menggunakan bantuan Ethernet Shield W5100 sebagai protokol penghubung ke jaringan.

D. Rumusan Masalah

Berdasarkan batasan masalah yang ada, maka dapat ditentukan rumusan permasalahan sebagai berikut:

1. Bagaimana rancang bangun dari *Prototype Urine Analizer Telemetry* Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang?
2. Bagaimana unjuk kerja dari *Prototype Urine Analizer Telemetry* Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang?.

E. Tujuan

Adapun tujuan yang ingin dicapai penulis dari pembuatan alat ini sebagai berikut:

1. Merancang alat *Prototype Urine Analyzer Telemetry* Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang.
2. Mengetahui unjuk kerja dari *Prototype Urine Analyzer Telemetry* Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang.

F. Manfaat

Pembuatan proyek akhir ini diharapkan dapat memberikan manfaat bagi mahasiswa, lembaga pendidikan, dan industri yaitu sebagai berikut:

1. Bagi Mahasiswa

Dapat menambah wawasan dan pengetahuan mahasiswa dalam bidang yang diteliti serta sebagai pengaplikasian ilmu yang telah didapatkan selama menempuh pendidikan di bangku perkuliahan.

2. Bagi Universitas

Dapat bermanfaat sebagai wujud dari perkembangan ilmu pengetahuan dan teknologi (IPTEK) serta aplikasi nyata pengembangan teknologi dalam dunia pendidikan, khususnya untuk Fakultas Teknik.

3. Bagi Masyarakat

Dapat diterapkan di pusat-pusat kesehatan masyarakat supaya pelayanan kesehatan masyarakat meningkat dan jumlah angka kematian serta dapat menjadi referensi teknologi alat medis selanjutnya.

G. Keaslian Gagasan

Pembuatan proyek akhir dengan judul “*Prototype Urine Analyzer Telemetry Menggunakan Sensor Warna untuk Mendeteksi Penyakit Diabetes pada Seseorang*” ini merupakan gabungan dan pengembangan dari ide-ide dasar yang bersifat *open source* yang telah dipublikasikan di internet maupun media cetak dengan gagasan pribadi yang terinspirasi dari kesulitan yang dialami oleh masyarakat. Meskipun kemungkinan terdapat kesamaan dengan yang sudah ada, namun penulis meyakini bahwa terdapat perbedaan baik secara teknik maupun konsep perancangan.

Berikut ini beberapa karya dan penelitian relevan yang dijadikan acuan dalam proyek akhir ini yaitu:

1. Rancang bangun alat ukur kadar gula darah non-invasive berbasis Mikrokontroler AT89S51 dengan mengukur tingkat kekeruhan spesimen urine menggunakan sensor potodioda (Eko Satria, Universitas Andalas, 2013). Tujuan dari karya ini yaitu membuat alat yang dapat mengukur kadar gula darah tanpa harus menguji dan memeriksa darah secara langsung dengan menggunakan parameter warna urine. Persamaan karya ini dengan penelitian yang akan dilakukan yaitu pada pemilihan urine sebagai indikator untuk menentukan kondisi tubuh seseorang. Sedangkan perbedaan karya ini dengan penelitian yang akan dilakukan adalah terletak pada penggunaan *chip* kontroler yang digunakan.
2. Rancang bangun alat deteksi dehidrasi menggunakan LED dan fotodioda melalui warna urine (Achmad Rokim, Universitas Islam Negeri Sunan Kalijaga Yogyakarta, 2015). Tujuan dari karya ini yaitu membuat alat yang dapat mendeteksi tingkat dehidrasi seseorang melalui tingkat warna urine. Persamaan karya ini dengan penelitian yang akan dilakukan yaitu pada pemilihan urine sebagai indikator.

Berdasarkan kedua penelitian di atas, keterkaitan antara karya-karya di atas dengan penelitian yang akan dilakukan ini adalah penggunaan warna urine sebagai indikator dalam menentukan kondisi seseorang apakah menderita penyakit diabetes ataupun tidak serta Arduino UNO digunakan sebagai kontroler alat yang akan dibuat karena *board* jenis ini selain memiliki harga yang relatif murah, juga dapat lebih mudah

digunakan. Sehingga hal ini diadaptasi dan diimplementasi dalam proyek akhir ini.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Kadar Gula Darah pada Diabetes Melitus

Kadar gula darah merupakan sejumlah glukosa yang terdapat di plasma darah (Dorland, 2010). Pemantauan kadar gula darah sangat dibutuhkan dalam menegakkan sebuah diagnosa terutama untuk penyakit diabetes melitus. Besarnya kadar gula darah seseorang dapat diperiksa secara langsung saat pasien datang untuk diperiksa ataupun saat sedang dalam kondisi puasa, namun besarnya berbeda yaitu saat pemeriksaan kadar glukosa darah sewaktu normalnya harus <200 mg/dL, sedang untuk hasil kadar glukosa saat puasa normalnya <126 mg/dL (Waspadji, 2007).

Glukosa merupakan pecahan dari karbohidrat yang akan diserap tubuh dalam aliran darah. Glukosa berperan sebagai bahan bakar utama dalam tubuh yang fungsinya menghasilkan energi (Amir, 2015). Glukosa darah dipengaruhi beberapa faktor, salah satunya yaitu usia (Syauqy, 2015). Semakin bertambahnya usia seseorang maka akan tinggi pula kemungkinan untuk menderita penyakit-penyakit, salah satunya diabetes melitus. Berdasarkan hasil penelitian, usia yang rentan terkena penyakit diabetes melitus adalah kelompok usia 45-54 tahun yang mana kelompok ini lebih tinggi 2,2% bila dibanding dengan kelompok usia 35-44 tahun (Fatimah, 2015).

Adapun kategori besaran jumlah glukosa darah manusia yang dapat dilihat pada Tabel 1 di bawah ini:

Tabel 1. Kategori glukosa darah

Kategori	Glukosa Puasa (mg/dL)
Normal	<100
Pra-diabetes	100 – 125
Diabetes	>126

(ADA, 2010)

Diabetes Mellitus (DM) merupakan penyakit metabolik kronik progresif yang ditandai dengan *hiperglikemi* (kadar gula darah tinggi) yang disebabkan oleh kurangnya sekresi insulin, aktivitas insulin, atau keduanya. Diabetes melitus adalah suatu penyakit kronis dimana terjadi akibat dari penurunan produksi insulin di pankreas. Penurunan tersebut terjadi karena tubuh tidak dapat merespon kerja insulin yang mana mengakibatkan insulin yang diproduksi tidak efektif sehingga sel-sel tubuh tidak dapat mengambil glukosa dan menggunakannya sebagai energi. Gejala khas yang sering timbul pada penderitanya yaitu seperti *polidipsia* (rasa haus berlebihan), *polyuria* (kencing yang berlebihan), *polyfagia* (rasa lapar yang berlebihan), penglihatan kabur, dan penurunan berat badan yang drastis.

Diabetes melitus merupakan penyakit metabolisme akibat adanya peningkatan kadar glukosa darah diatas normal. Penderita diabetes melitus akan merasakan beberapa gejala seperti sering haus, mudah lapar, dan sering buang air kecil dengan jumlah diatas normal. Urin pada penderita diabetes melitus pada umumnya lebih transparan atau tidak memiliki

warna sama sekali dan beraroma manis, sehingga beberapa orang menyebutnya dengan kencing manis.

Saat ini penyakit diabetes melitus dapat diklasifikasi menjadi tiga tipe yaitu diabetes tipe 1, tipe 2, dan tipe gestasional. Diabetes tipe 1 adalah penyakit yang disebabkan karena gangguan automun, dimana sel beta pankreas pada telah mengalami kerusakan. Tipe ini umumnya menyerang anak di bawah usia 35 tahun (Sari, 2012). Sedangkan diabetes tipe 2 disebabkan karena gangguan sekresi insulin dan resistensi terhadap insulin. Tipe ini umumnya terjadi di usia 30 tahun ke atas dan timbul secara perlahan. Selanjutnya diabetes tipe gestasional yang mana tipe ini terjadi selama kehamilan karena disebabkan oleh perubahan reabsorpsi makanan. Oleh sebab itu, pada proyek akhir ini penderita diabetes tipe 2 menjadi objek pengukuran. Hal ini disebabkan tipe ini dapat digolongkan tipe gawat dan diperlukan penanganan yang khusus karena timbulnya secara perlahan.

Kebanyakan penyakit diabetes melitus belum diketahui penyebab utamanya. Terdapat dua faktor yang menyebabkan terjadinya DM yaitu faktor yang tidak dapat dimodifikasi (ras, usia, jenis kelamin, dan riwayat keluarga dengan DM) dan dapat dimodifikasi (pola hidup, obesitas, aktivitas). Terdapat lima langkah untuk mendiagnosis diabetes melitus, salah satunya dengan cara tes urine. Tes urine digunakan untuk mengetahui kandungan gula di dalam urine. Tes ini meliputi uji Benedict dan uji Dipstick.

Uji Benedict digunakan untuk menentukan adanya glikogen dalam urine. Mula-mula sampel urine dari penderita diabetes diambil. Kemudian diambil 8 tetes urine tersebut ke dalam tabung reaksi. Selanjutnya sampel tersebut ditetesi dengan pereaksi Benedict sebanyak 5 tetes. Kemudian sampel tersebut dipanaskan sampai terbentuk warna. Sifat warna inilah yang memberikan petunjuk kadar gula dalam urine. Pada hasil uji Benedict, jika warna yang dihasilkan adalah merah bata maka urine tersebut mengandung lebih dari 2% glukosa yang artinya orang tersebut menderita penyakit diabetes.

Pada dasarnya uji Benedict dilakukan untuk mengetahui kandungan senyawa aldehida. Oleh karena itu, pada uji Benedict akan memberikan warna bahkan jika ada gula-gula lain yang terdapat dalam urine, seperti maltosa, galaktosa, sukrosa, fruktosa, dan lain-lain. Namun uji Benedict tidak dapat digunakan untuk penderita hipoglikemia. Sedangkan pada uji Dipstick digunakan untuk memastikan adanya gula dalam urine. Pada dasarnya Dipsticks merupakan strip kertas yang mengandung zat kimia tertentu dan akan berubah warna jika bereaksi dengan gula. Perubahan warna yang terjadi tergantung pada bahan kimia yang digunakan dalam pembuatan dipstick tersebut. Pada uji Dipstick warna yang dihasilkan kemudian dibandingkan dengan warna yang terdapat pada buku manual.

Pada dasarnya diabetes melitus tipe 2 dapat dilakukan pencegahan secara bertahap. Menurut (Hasnah, 2009), terdapat empat tingkatan pencegahan pada penderita diabetes melitus tipe 2 yaitu:

1. Pencegahan Tingkat Dasar (*Primordial Prevention*)

Pencegahan tingkat dasar ini diupayakan mempertahankan keadaan resiko rendah terhadap penyakit secara umum, yaitu meliputi usaha memelihara dan mempertahankan kebiasaan atau perilaku hidup sehat misalnya dengan menjaga pola makan sehat dan kebiasaan berolahraga.

2. Pencegahan Tingkat Pertama (*Primary Prevention*)

Pencegahan tingkat pertama diupayakan agar tidak timbul penyakit DM yang disebabkan oleh faktor keturunan dan lingkungan (faktor jasmani yang kurang sehat, obesitas, nutrisi berlebih, dan obat-obatan).

3. Pencegahan Tingkat Kedua (*Secondary Prevention*)

Sasaran utama ditujukan pada penderita yang baru saja terkena penyakit DM dengan melalui diagnosa dini dan pengobatan yang cepat dan tepat, kegiatan ini melalui *screening*.

4. Pencegahan Tingkat Ketiga (*Tertiary Prevention*)

Pencegahan ini merupakan pencegahan dengan sasaran utamanya yaitu mencegah bertambahnya beratnya penyakit atau mencegah kecacatan maupun kematian, kecacatan yang mungkin timbul akibat DM seperti pembuluh darah otak bisa menyebabkan *stroke*, pembuluh darah mata menyebabkan kebutaan, dan pembuluh darah ginjal menyebabkan gagal ginjal.

B. Urine Analysis

Urine merupakan sebuah limbah cair yang diekskresi atau dikeluarkan dari tubuh melalui ginjal dalam proses urinisasi dan dikeluarkan melalui uretra. Di dalam urine terkandung hasil serangkaian proses metabolisme seluler, dimana paling banyak nitrogen yang harus dikeluarkan dari aliran darah. Normalnya, urine mengandung beberapa zat seperti air, urea, asam urat, amoniak, kreatinin, asam laktat, asam fosfat, asam sulfat, klorida, dan beberapa zat yang berlebihan dalam darah misalnya vitamin C atau obat-obatan. Proses mendeteksi bahan-bahan kimia yang terkandung dalam urine tersebut dinamakan urinalisis.

Urine merupakan salah satu cairan tubuh yang dapat digunakan untuk menganalisa kesehatan seseorang dengan melalui serangkaian pemeriksaan sehingga dapat didiagnosa suatu penyakit melalui hasil pemeriksaan tersebut. Pemeriksaan urine rutin umumnya terdiri dari pengukuran jumlah urine, warna, kejernihan, berat jenis, pH, sedimen urine. Pemeriksaan kandungan urine umumnya disebut juga urinalisis. Jumlah urine normal pada usia lanjut rata-rata berkisar 1-2 liter setiap harinya, namun jumlah tersebut juga dapat berubah sesuai dengan jumlah cairan yang masuk ke tubuh setiap harinya.

Adapun jumlah urine seseorang sesuai usia dapat dilihat pada Tabel 2 sebagai berikut:

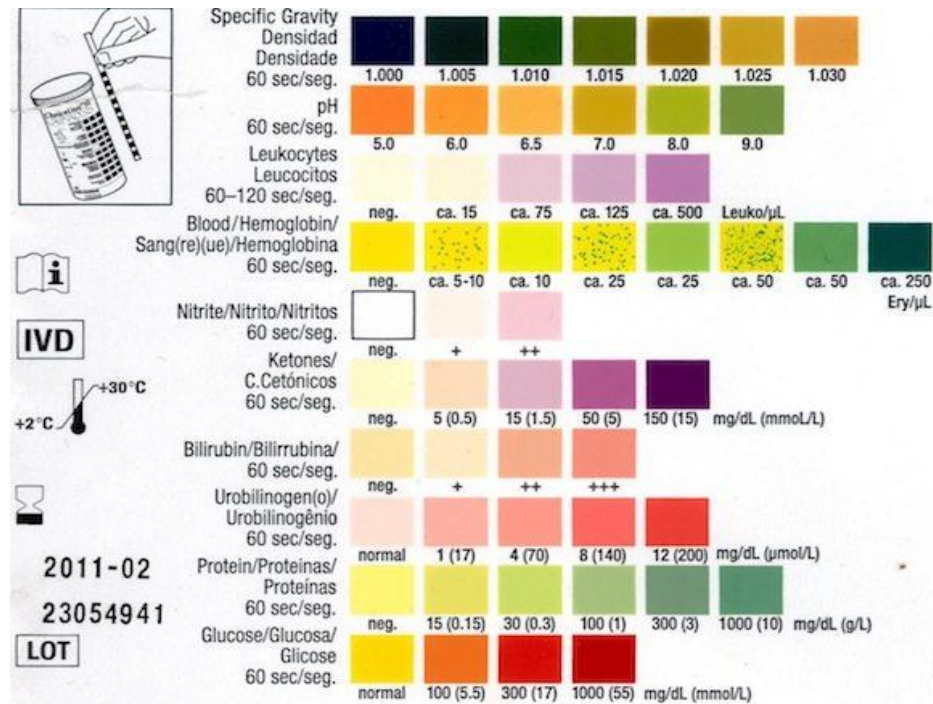
Tabel 2. Jumlah urine berdasarkan usia

No.	Usia	Jumlah/hari
1.	1-2 hari	15-60ml
2.	3-10 hari	100-300 ml
3.	10 hari-2 bulan	250-400 ml
4.	2 bulan-1 tahun	400-500 ml
5.	1-3 tahun	500-600 ml
6.	3-5 tahun	600-700 ml
7.	5-8 tahun	700-1000 ml
8.	8-14 tahun	800-1400 ml
9.	14 tahun-dewasa	1500 ml
10.	Dewasa tua	≤ 1500 ml

(A. Aziz Alimul Hidayat, 2006)

Sementara pemeriksaan warna urine dilakukan dengan menggunakan tes urine strip atau metode *light microscopy*. Melalui metode ini, seseorang dapat diketahui menderita penyakit seperti fungsi ginjal, saluran kemih, pankreas dan hati. Tes ini dapat dilakukan secara manual ataupun otomatis. Perbedaan urine strip secara manual dan otomatis yaitu pada pembacaan perubahan warna setiap pad pada urine strip yang tercelup urine. Apabila secara manual, perubahan warna yang terlihat secara visual pada *pad* yang berubah warna menunjukkan satu parameter untuk dicocokkan dengan tabel yang tersedia (diberikan oleh *manufacture*). Sedangkan secara otomatis, analisa menggunakan alat urine *analyzer* yang dapat mempermudah menentukan jumlah (kuantifikasi) analit dalam urine, termasuk didalamnya bilirubin, protein, glukosa, dan sel darah.

Berikut ini gambar bagan warna pada *pad urine strip* yang dihasilkan saat melakukan tes urine yang dapat dilihat pada Gambar 1.



Gambar 1. Bagan warna Dipstick Urinalysis

(https://lifeinthefastlane.com/wp-content/uploads/2010/04/Urinalysis_2.jpg)



Gambar 2. Contoh warna urine

(<https://lifeinthefastlane.com/investigations/urine-colour/>)

Pada orang normal, warna urine seseorang akan cenderung berwarna jernih transparan dengan sedikit kuning karena pengaruh warna empedu. Namun ada kalanya warna urine seseorang akan sedikit berbeda apabila

seseorang sedang mengkonsumsi makanan atau obat-obatan tertentu. Hal tersebut tidak masalah karena akan warna urine kembali normal. Tetapi jika warna urine yang dianggap tidak normal ini berlangsung secara terus menerus bisa jadi indikator dari keberadaan penyakit. Beberapa penyakit yang dapat dideteksi melalui urine diuraikan diantaranya yaitu ginjal, diabetes melitus, hepatitis B, infeksi saluran kemih, dan kanker prostat. Penderita diabetes melitus akan merasakan beberapa gejala seperti sering haus, mudah lapar, dan sering buang air kecil dengan jumlah diatas normal. Urin pada penderita diabetes melitus pada umumnya lebih transparan atau tidak memiliki warna sama sekali dan beraroma manis, sehingga beberapa orang menyebutnya dengan kencing manis.

Berikut beberapa warna urine yang dapat menjadi indikasi adanya gangguan kesehatanyang mana dapat dilihat pada Tabel 3 dibawah ini:

Tabel 3. Interpretasi warna urine

No.	Keadaan	Normal	Interpretasi
1.	Warna	Kekuning-kuningan	Urine berwarna oranye gelap menunjukkan adanya pengaruh obat, sedangkan warna merah dan kuning kecoklatan mengindikasikan adanya penyakit.
2.	Bau	Aromatik	Bau menyengat merupakan indikasi adanya masalah seperti infeksi atau penggunaan obat tertentu.
3.	Berat jenis	1,01-1,03	Menunjukkan adanya konsentrasi urine.
4.	Kejernihan	Terang dan transparan	Adanya kekeruhan karena mukus atau pus.
5.	pH	Sedikit asam (4,5-7,5)	Dapat menunjukkan keseimbangan asam-basa; bila bersifat alkali menunjukkan

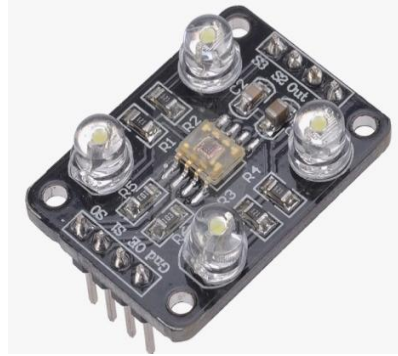
			adanya aktivitas bakteri.
6.	Protein	Molekul protein yang besar seperti: albumin, fibrinogen, atau globulin tidak dapat disaring melalui ginjal	Pada kondisi kerusakan ginjal, molekul tersebut dapat melewati saringan masuk ke urine.
7.	Darah	Tak tampak jelas	Hematuria menunjukkan trauma atau penyakit pada saluran kemih bagian bawah.
8.	Glukosa	Adanya sejumlah glukosa dalam urine	Apabila frekuensinya terus menerus dapat diduga menderita penyakit diabetes melitus.

(A. Aziz Alimul Hidayat, 2006)

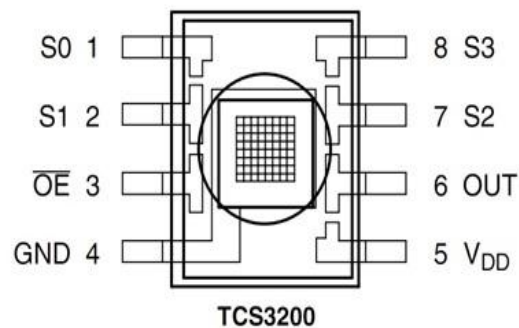
C. TCS230

Sensor TCS230 adalah sensor warna yang umumnya banyak dimanfaatkan sebagai pendeteksi objek benda berupa warna. Sensor ini juga dapat digunakan sebagai sensor gerak, dimana sensor dapat mendeteksi gerakan suatu objek berdasarkan perubahan warna yang diterima oleh sensor. Pada dasarnya, sensor ini merupakan rangkaian photodiode yang disusun secara matriks array 8x8 dengan terdapat 16 buah konfigurasi photodiode yang berfungsi sebagai filter warna merah, 16 photodiode sebagai filter warna biru dan 16 photodiode lagi tanpa filter warna. Sensor warna TCS230 dikemas dalam *chip* DIP 8 pin dengan bagian depan komponen transparan sebagai tempat menerima intensitas cahaya yang berwarna.

Berikut ini bentuk fisik dari sensor warna TCS230 dapat dilihat pada Gambar 3 di bawah ini:



Gambar 3. Bentuk fisik TCS230
(<http://www.arduinoocia.com>)



Gambar 4. Konstruksi TCS230
(Elektronika Dasar, 2012)

Adapun fungsi pada masing-masing pin dapat dilihat pada Tabel 4 sebagai berikut:

Tabel 4. Fungsi pin sensor warna TCS230

Nama	Pin	I/O	Fungsi Pin
<i>GND</i>	4	-	Sebagai <i>ground</i> pada <i>power supply</i>
OE	3	I	<i>Output enable</i> , sebagai <i>input</i> untuk frekuensi <i>output</i> skala rendah
<i>OUT</i>	6	O	Sebagai <i>output</i> frekuensi
S0, S1	1, 2	I	Sebagai saklar pemilih pada frekuensi <i>output</i> skala tinggi
S2, S3	7, 8	I	Sebagai saklar pemilih 4 kelompok dioda
V _{DD}	5	-	Suplai tegangan

(Elektronika Dasar, 2012)

Sensor warna TCS230 bekerja dengan cara membaca nilai intensitas cahaya yang dipancarkan oleh led *super bright* terhadap objek, pembacaan nilai intensitas cahaya tersebut dilakukan melalui matrik 8x8 photodiode, dimana 64 photodiode tersebut terbagi menjadi 4 kelompok pembacaan warna, setiap warna yang disinari led akan memantulkan sinar led menuju photodiode, pantulan sinar tersebut memiliki panjang gelombang yang berbeda-beda tergantung pada warna objek yang terdeteksi. Hal ini yang membuat sensor warna TCS230 dapat membaca beberapa macam warna.

Panjang gelombang dan sinar led yang dipantulkan objek berwarna berfungsi mengaktifkan salah satu kelompok photodiode pada sensor warna tersebut, sehingga ketika kelompok photodiode yang digunakan telah aktif, S2 dan S3 akan mengirimkan sinyal ke mikrokontroler untuk menginformasikan warna yang dideteksi. Berikut ini Tabel 5 yang menampilkan pemilihan mode pengelompokkan photodiode pembaca warna.

Tabel 5. Mode pemilihan photodiode pembacaan warna

S2	S3	Photodiode
0	0	Merah
0	1	Biru
1	0	<i>Clear (No Filter)</i>
1	1	Hijau

(Palensyah, 2014)

D. Arduino

Arduino adalah sebuah *platform* dari *physical computing* yang bersifat *open source* (Aslamia, 2015). Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi merupakan kombinasi dari *hardware*, bahasa

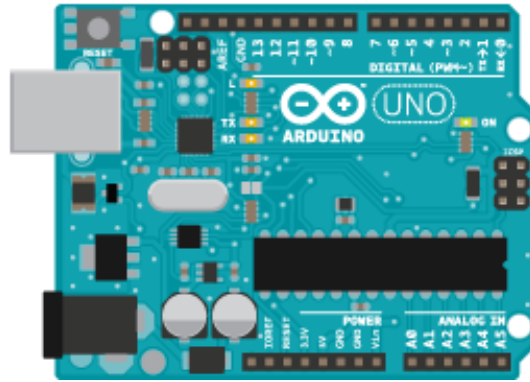
pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan yang digunakan untuk menulis program, meng-*compile* menjadi kode biner dan mengunggahnya ke dalam memori mikrokontroler. Menurut (Syahwil, 2013), Arduino adalah papan elektronik *open source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler AVR dari perusahaan Atmel. Berikut ini karakteristik dan struktur arduino sebagai berikut:

1. Arduino IDE merupakan multi *platform* yang dapat dijalankan pada berbagai sistem operasi seperti *Window* sataupun *Linux*. IDE adalah program komputer yang berfungsi untuk menyediakan semua fasilitas yang diperlukan dalam pembangunan perangkat lunak baik *editor*, *compiler*, *linker*, maupun *debugger*.
2. Pemrograman pada *hardware* Arduino menggunakan kabel yang terhubung dengan *port Universal Serial Bus* (USB). Hal ini disebabkan karena banyak komputer sekarang yang tidak dilengkapi dengan *port* serial.
3. Arduino adalah *hardware* dan *software* yang bersifat *open source* yaitu sistem pengembangan yang tidak dikoordinasi oleh individu atau lembaga pusat, namun oleh para pelaku yang bekerjasama dengan memanfaatkan kode sumber (*source code*).
4. Biaya pembelian *hardware* tergolong cukup murah sehingga tidak akan menghabiskan banyak uang untuk memilikinya.

1. Arduino Uno Rev 3

a. Arsitektur Arduino Uno

Arduino uno adalah *board* yang merupakan pengembangan dari mikrokontroler ATmega328. Dengan memiliki 14 pin digital untuk input atau output digital. *Board* ini menggunakan Kristal 16MHz, pin ICSP dan tombol reset, serta *input* dapat menggunakan dari USB langsung maupun dari tegangan DC 7-12V DC.



Gambar 5. Arduino Uno R3
(www.arduino.cc)

Adapun spesifikasi dari *board* Arduino dapat dilihat pada Tabel 6 di bawah ini.

Tabel 6. Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Operasi Tegangan	5 Volt
<i>Input</i> Tegangan	7-12 Volt
Pin I/O Digital	14
Pin Analog	6
Arus DC tiap pin I/O	50 mA
Arus DC ketika 3.3V	50 mA
Memori Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Kecepatan Clock	16 MHz

(Wicaksono, 2017)

b. ATmega 328p

ATmega328 merupakan sebuah mikrokontroler keluarga 8 bit. ATmega328 memiliki 3 buah PORT utama yaitu PORTB, PORTC, dan PORTD dengan total memiliki pin I/O sebanyak 23 pin. Port B merupakan jalur data 8 bit yang dapat difungsikan sebagai I/O. Port B juga mempunyai fungsi lainnya yaitu: 1) ICP1 (PB0),

berfungsi sebagai *timer counter* 1 *inputcapture* pin, 2) OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM, 3) MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI, 4) Selain itu pin ini juga berfungsi sebagai jalur pemograman serial (ISP), 5) TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*, dan 6) XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

Sementara Port C merupakan jalur data 8 bit yang dapat difungsikan sebagai pin *input/output* digital. Fungsi lain PORTC antara lain yaitu : 1) ADC6 *channel* (PC0 - PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital, 2) I2C (SDA dan SDL) yang digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I2C.

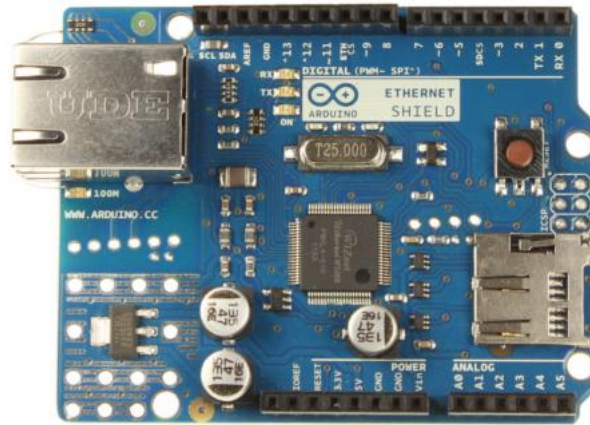
Sedangkan Port D merupakan jalur data 8 bit yang dapat difungsikan sebagai pin *input/output*. Port D juga mempunyai fungsi lain yaitu: 1) USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL, 2) *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, 3) XCK berfungsi sebagai sumber *clock* eksternal untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU,

sehingga tidak perlu membutuhkan *external clock*, 4) T0 dan T1 berfungsi sebagai masukan *counter* eksternal untuk *timer* 1 dan *timer* 0, dan 5) AIN0 dan AIN1 keduanya merupakan masukan untuk analog komparator.

2. Ethernet Shield W5100

Secara dasarnya *shield* adalah papan rangkaian yang belum disempurnakan atau dilengkapi secara keseluruhan dan dibuat dengan pin yang sesuai dengan papan Arduino agar mudah saat digunakan atau dihubungkan. Arduino Ethernet Shield merupakan *board* yang digunakan untuk menghubungkan Arduino agar dapat terkoneksi dengan jaringan komputer, dengan menggunakan *ethernet chip* Wiznet W5100 yang menyediakan layanan jaringan dengan kemampuan TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*). Pada *board* ini juga dilengkapi dengan *SD card slot* yang dapat digunakan untuk menyimpan beberapa format file untuk diakses oleh Arduino. *Board* ini dipasang di atas *board* Arduino yang lain yang mana Arduino Ethernet Shield ini akan menjadi perantara atau penghubung Arduino dengan PC, switch, atau router dengan menggunakan kabel LAN (*Local Area Network*) yang memiliki konektor RJ45 sehingga memungkinkan Arduino dapat terhubung dengan suatu jaringan.

Adapun bentuk fisik dari Arduino Ethernet Shield dapat dilihat pada Gambar 6 di bawah ini:



Gambar 6. Bentuk fisik Arduino Ethernet Shield

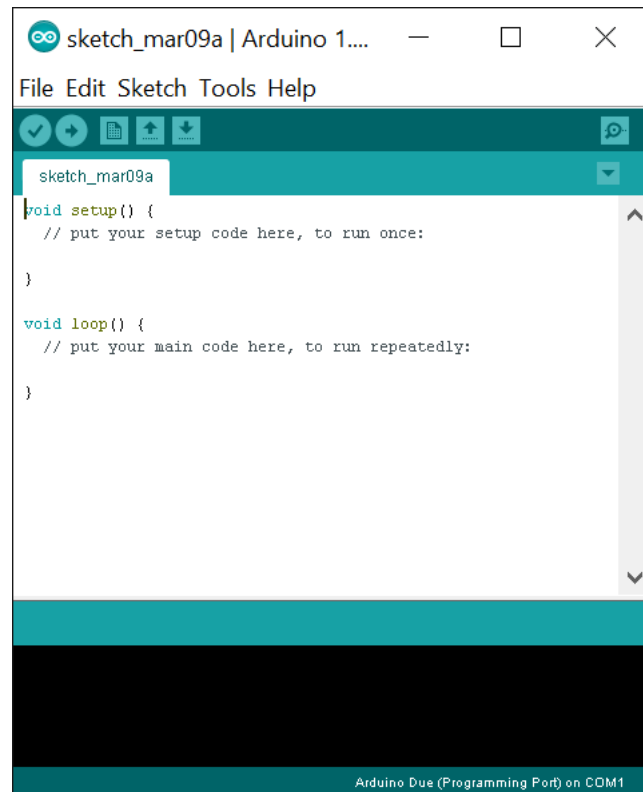
(www.arduino.cc/)

Perantara atau penghubung perangkat Arduino dengan jaringan internet dipengaruhi salah satunya dengan *MAC Address* (*Media Access Control Address*). *MAC Address* adalah alamat fisik suatu *interface* jaringan yang bersifat unik dan berfungsi sebagai identitas dari perangkat tersebut. Secara umum format *MAC Address* terdiri dari 6 segmen dengan masing-masing 2 digit heksadesimal, dan setiap segmen dipisahkan oleh tanda (-) atau (:) dengan tiga segmen pertama merupakan kode vendor dan 3 segmen berikutnya merupakan *serial number* kartu tersebut. Sedangkan pada produk tiruan tidak disertai *MAC Address* pada Arduino Ethernet Shield, namun tetap dapat digunakan hanya pada jaringan lokal saja, dengan menggunakan *MAC Address default* yang dapat dipakai pada jaringan *localhost*. Sebagai contohnya *MAC Address default* yang dapat digunakan pada *library* Arduino Ethernet yaitu, DE : AD : BE : EF : FE : ED.

Pada tugas akhir ini, penulis menggunakan Arduino Ethernet Shield untuk menghubungkan Arduino dengan *client* karena praktis, sederhana, dan juga dapat memungkinkan terhubung dengan jaringan komputer dalam hitungan detik. Sehingga hal ini dipilih untuk digunakan sebagai penghubung perangkat Arduino ke jaringan.

3. Arduino IDE

Arduino IDE adalah *software* yang dirancang menggunakan bahasa pemrograman Java yang terdiri dari 3 bagian yaitu *Editor* program, *Compiler*, dan *Uploader*. Pada bagian *editor*, *user* dapat melakukan pemrograman baik itu menulis ataupun mengedit program dalam bahasa *processing*. Sementara pada bagian *Compiler* berisi pengubah kode program menjadi kode biner agar kode tersebut dapat terbaca oleh mikrokontroler. Sedangkan bagian *Uploader* digunakan untuk menuliskan dan menyalin kode biner dari komputer ke dalam memori *board* Arduino. Bahasa pemrograman yang digunakan pada Arduino IDE adalah bahasa C yang telah disederhanakan, yaitu telah disediakan berbagai library pada Arduino untuk mempermudah pemrograman.



Gambar 7. Tampilan Arduino IDE

(<https://blog.arduino.cc/>)

Pada tampilan Arduino IDE terdapat *toolbar* yang didesain untuk mempermudah dalam melakukan pemrograman. Berikut ini fungsi-fungsi pada *toolbar IDE* sebagai berikut:

- a. *Verify*, digunakan untuk melakukan kompilasi program yang saat di *editor*.
- b. *New*, digunakan untuk membuat program baru dengan mengosongkan isi dari jendela *editor* saat ini.
- c. *Open*, digunakan untuk membuka program yang ada dari sistem file.

- d. *Save*, digunakan untuk menyimpan program saat ini.
- e. *Upload*, digunakan untuk menyalin data hasil pemrograman dari komputer ke dalam *memory board* Arduino. Ketika melakukan *upload*, maka harus melakukan pengaturan jenis Arduino dan *port* COM yang digunakan.
- f. *Serial Monitor*, digunakan untuk melihat hasil pemrograman yang telah tersimpan dalam *memory* Arduino.

Kode program yang dituliskan pada *chip* mikrokontroler Arduino umumnya menggunakan beberapa fungsi seperti tipe data, operator, dan program kontrol. Pada setiap bagian dari data yang disimpan dalam program Arduino umumnya memiliki tipe datanya masing-masing. Berikut ini beberapa tipe data yang terdapat pada Arduino yang ditunjukkan pada Tabel 7

Tabel 7. Tipe data bahasa C

Jenis	Range
<i>int / sign int</i>	-32768 - +32767 ($2^{15} - 1$)
<i>unsign int</i>	0 – 65535
<i>short int / signed short int</i>	-28 – 127
<i>unsigned short int</i>	0 – 255
<i>long int / signed long int</i>	-2147483648 – 2147483648
<i>unsigned long int</i>	0 – 4294967296
<i>Char</i>	karakter ASCII
<i>unsigned char</i>	0 – 255
<i>signed char</i>	-128 - 127
<i>float</i>	maksimum nilai 6 digit
<i>double</i>	maksimum nilai 12 digit
<i>long double</i>	maksimum nilai 24 digit

(Widodo, 2016)

Sementara operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai ke suatu variabel, membandingkan kesamaan dua buah nilai. Sebagian operator C tergolong sebagai operator binari, yaitu operator yang dikenakan terhadap dua buah nilai (*operand*). Berikut ini beberapa operasi kondisi yang sering digunakan dalam bahasa C yang dapat dilihat pada Tabel 8 sebagai berikut:

Tabel 8. Operasi kondisi

Operator	Keterangan
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar atau sama dengan
=	Sama dengan
!=	Tidak sama dengan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagi (modulus)
!	<i>Boolean</i> NOT
&&	<i>Boolean</i> AND
	<i>Boolean</i> OR
~	Komplemen <i>Bitwise</i>
&	<i>Bitwise</i> AND
	<i>Bitwise</i> OR
^	<i>Bitwise Exclusive</i> OR
>>	<i>Right Shift</i>
<<	<i>Left Shift</i>
=	Untuk memasukkan nilai
+=	Untuk menambah nilai dari keadaan semula
-=	Untuk mengurangi nilai dari keadaan semula
*=	Untuk mengalikan nilai dari keadaan semula
/=	Untuk melakukan pembagian terhadap bilangan semula
%=	Untuk memasukkan nilai sisa bagi dari

	pembagian bilangan semula
<code><< =</code>	Untuk memasukan <i>shift left</i>
<code>>> =</code>	Untuk memasukan <i>shift right</i>
<code>& =</code>	Untuk memasukan <i>bitwise AND</i>
<code>^ =</code>	Untuk memasukan <i>bitwise XOR</i>
<code>\ =</code>	Untuk memasukan <i>bitwise OR</i>

(Widodo, 2016)

Selain itu, dalam melakukan pemrograman menggunakan Arduino IDE juga dibutuhkan program kontrol agar program yang akan dibuat sesuai dengan tujuan. Berikut ini program kontrol dalam bahasa C sebagai berikut:

a. Percabangan *if* dan *if ... else ...*

Perintah *if* dan *if ... else ...* digunakan untuk melakukan operasi percabangan bersyarat.

b. Percabangan *switch*

Pernyataan *switch* adalah sebuah variabel secara berurutan diuji oleh beberapa konstanta bilangan bulat atau konstanta karakter sintaks perintah *switch*.

c. Looping

Looping adalah pengulangan satu atau beberapa perintah sampai mencapai keadaan tertentu. Ada tiga perintah *looping*, yaitu:

1) *for ...*

2) *while ...*

3) *do...while....*

E. Notepad++

Notepad++ adalah standar *programmer editor*. Notepad++ adalah sebuah text editor yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan penyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan di atas sistem operasi *Microsoft Windows*. Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad++ juga dilisensikan sebagai perangkat *free*. Beberapa daftar bahasa program yang didukung oleh Notepad++ adalah C, C++, Java, C#, XML, HTML, PHP, Javascript,.

Pada pembuatan *prototype urine analyzer telemetry* ini, Notepad++ digunakan untuk membuat program tampilan *web browser* dengan format ekstensi .htm atau .html. Notepad++ juga dapat digunakan sebagai *text editor HTML (Hyper Text Markup Language)* untuk membuat kode HTML. Pada dasarnya HTML hanya *text* biasa yang ditulis dalam kode-kode khusus (*script*). *Web browser* lah yang akan menerjemahkan kode HTML ini menjadi tampilan halaman *web* atau sebagai *compiler* kode tersebut. Setiap halaman HTML harus disimpan dengan ekstensi *.html. Namun beberapa halaman *web* juga memiliki ekstensi *.htm, ekstensi ini digunakan untuk mendukung Windows versi lama yang masih menggunakan tiga huruf belakang sebuah file, begitu halnya Arduino hanya mengenali format file dengan tiga huruf untuk penyimpanan SD *card*.

HTML (*Hyper Text Markup Language*) adalah bahasa pemrograman (*script*) yang digunakan untuk menyusun dokumen-dokumen *web*. Dokumen HTML disimpan dalam format teks reguler (berupa file *.html atau *.htm) dan mengandung tag-tag yang memerintahkan *web browser* untuk mengeksekusi perintah-perintah yang dispesifikasikan. HTML tidak membedakan huruf besar dan huruf kecil pada penulisan elemen maupun tag. Penulisan `<i>` dan `<I>` dianggap sama. Beberapa *browser* ada yang tidak dapat mengenali tag tertentu, dan *browser* tersebut akan mengabaikan tag yang tidak dikenalnya lalu menuliskan isi di dalam tag tersebut sebagai teks biasa.

Struktur dasar dokumen HTML adalah sebagai berikut:

```
<!DOCTYPE html>
<html>
  <head>
    <title> ... </title>
  </head>
  <body> ...
  </body>
</html>
```

Dokumen HTML dibagi ke dalam tiga bagian, yaitu:

1. DOCTYPE atau DTD

DOCTYPE atau DTD (Document Type Declaration)

2. Tag `<html>`

Setiap dokumen HTML harus diawali dan ditutup dengan tag HTML.

Dengan sintak:

```
<HTML> ... </HTML>
```

Tag HTML akan memberitahu *browser* bahwa yang ada di dalam kedua tag tersebut adalah dokumen HTML.

- a. Tag adalah teks khusus (*markup*) berupa duakarakter “<” dan “>”.

Secara umum tag ditulis secara berpasangan, yang terdiri atas tag pembuka dan tag penutup (ditambah karakter “/” setelah “<”).

- b. Elemen terdiri atas tiga bagian, yaitu tag pembuka, isi dan tag penutup. Sebagai contoh untuk menampilkan judul dokumen HTML pada *browser* digunakan Elemen TITLE. Dengan sintak:

```
<TITLE><!--tag pembuka judul dokumen HTML-->
  Welcome to HTML <!--isi judul dokumen HTML-->
</TITLE><!-- tag penutup judul dokumen HTML-->
```

Penulisan program yang baik adalah secara struktur. Tag-tag ditulis secara berpasangan pada suatu elemen HTML, tidak boleh saling tumpang tindih dengan tag-tag lainnya.

- c. Atribut, digunakan untuk mendefinisikan properti dari suatu elemen HTML yang terdiri atas nama dan nilai.

```
<TAG>
  Nama Attribute="Nilai Attribut"
</TAG>
```

Secara umum atribut harus dalam tanda petik satu (‘) atau (“)

3. Tag <head>

Bagian header dari dokumen HTML di apit oleh tag HEAD. Dengan sintak:

```
<head> ... </head>
```

Di dalam ini juga diletakan tag TITLE yang menampilkan judul dari halaman pada browser, dan dipakai sebagai kunci pencarian saat client menggunakan fasilitas pencarian (search) pada browser. Dengan sintak:

```
<title> ... </title>
```

Header ini juga memuat tag META

4. Tag <body>

Elemen ini digunakan untuk menampilkan isi dokumen HTML yang akan ditampilkan pada website, baik berupa teks, gambar, sound, video, dan yang lainnya. Dengan sintak:

```
<body> ... </body>
```

Di dalam tag *body*, semua pengaturan format tampilan dilakukan, seperti menentukan warna latar , warna teks, warna dari link aktif, menentukan batas kiri dokumen, menentukan jenis font dari teks, membuat baris baru, dan lain-lain.

F. Catu Daya

Regulator adalah rangkaian pengatur tegangan keluaran, untuk mempertahankan kesetabilan level tegangan. Secara otomatis rangkaian regulator difungsikan untuk memastikan mempertahankan tegangan pada level tertentu, dimana tegangan *output* DC tidak dipengaruhi oleh perubahan tegangan *input*, suhu ataupun beban pada *output*.

Pada rangkaian elektronik yang bersifat digital seperti mikrokontroler dibutuhkan tegangan yang setabil dan bebas dari gangguan *noise* ataupun fluktuasi (naik turun) tegangan.

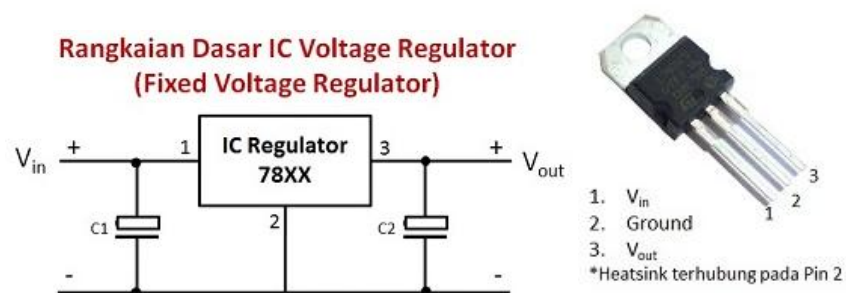
Salah satu tipe IC penstabil tegangan yang paling sering ditemukan adalah tipe 7805 yaitu IC *voltage regulator* yang mengatur tegangan *output* stabil pada tegangan 5 Volt DC. IC ini termasuk jenis penstabil

tegangan tetap (*Fixed Voltage Regulator*), yaitu memiliki nilai tegangan yang tetap atau tidak dapat disetel (di-*adjust*) sesuai dengan keinginan rangkaiannya. Tegangannya telah ditetapkan oleh produsen IC sehingga tegangan DC yang diatur juga tetap sesuai dengan spesifikasi IC-nya. Misalnya IC *voltage regulator* 7808, maka *output* tegangan DC-nya juga hanya 8 Volt DC. Terdapat 2 jenis pengatur tegangan tetap yaitu *positive voltage regulator* dan *negative voltage regulator*.

Jenis IC *voltage regulator* yang paling sering ditemukan di pasaran adalah tipe 78XX. Tanda XX dibelakangnya adalah kode angka yang menunjukkan tegangan *output* DC pada IC *voltage regulator* tersebut. Contohnya 7805, 7808, 7812 dan lain sebagainya. IC 78XX merupakan IC jenis *Positive Voltage Regulator*.

IC yang berjenis *negative voltage regulator* memiliki desain, konstruksi dan cara kerja yang sama dengan jenis *positive voltage regulator*, yang membedakannya hanya polaritas pada Tegangan Outputnya. Contoh IC jenis *negative voltage regulator* diantaranya adalah 7905, 7908, 7912 atau IC *voltage regulator* berawalan kode 79XX.

IC *Fixed Voltage Regulator* juga dikategorikan sebagai IC *Linear Voltage Regulator*. Dibawah ini adalah Rangkaian Dasar untuk IC LM78XX beserta bentuk Komponennya (*Fixed Voltage Regulator*). (Dickson Kho, 2017)



Gambar 8. Rangkaian dan IC regulator

(<http://teknikelektronika.com/wp-content/uploads/2014/10/Rangkaian-IC-Fixed-Voltage-Regulator.jpg?x22079>)

Adapun karakteristik dari IC regulator yang digunakan dalam catu daya yang dapat dilihat pada Tabel 9.

Tabel 9. Karakteristik regulator tegangan seri 78XX

Tipe	V-out (V)	I-out (A)			V-in (V)	
		78xx	78Lxx	78Mxx	Min	Max
7805	5	1	0,1	0,5	7,5	20
7806	6	1	0,1	0,5	8,6	21
7808	8	1	0,1	0,5	10,6	23
7809	9	1	0,1	0,5	11,7	24
7810	10	1	0,1	0,5	12,7	25
7812	12	1	0,1	0,5	14,8	27
7815	15	1	0,1	0,5	18	30
7818	18	1	0,1	0,5	21	33

(Elektronika Dasar, 2012)

BAB III

KONSEP RANCANGAN

A. Identifikasi Kebutuhan

Untuk merancang alat *prototype urine analyzer telemetry*, diperlukan beberapa komponen yaitu:

1. dibutuhkan sensor warna untuk mendeteksi warna urine,
2. komponen pengendali mikrokontroler yang akan mengolah data sensor berdasarkan keadaan urine,
3. media komunikasi protokol LAN untuk menghubungkan perangkat ke jaringan komputer agar dapat di akses oleh lebih dari satu *client* dan dari jarak jauh,
4. media penampil untuk menampilkan data dari urine yang telah diidentifikasi,
5. diperlukan *SD card* untuk menyimpan file halaman *web*,
6. power supply sebagai sumber tegangan.

B. Analisis Kebutuhan

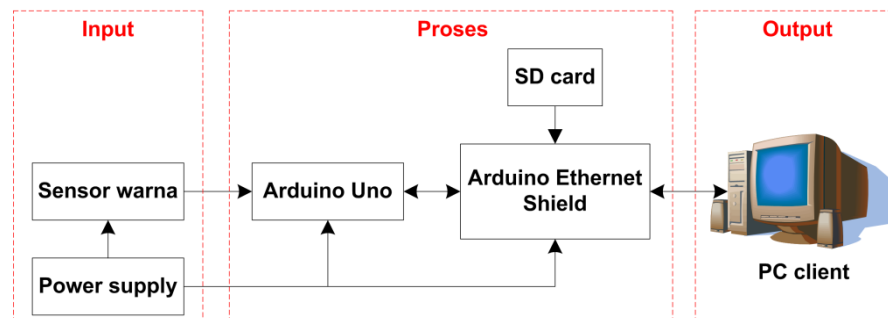
Berdasarkan identifikasi kebutuhan di atas, maka diperoleh beberapa analisis kebutuhan terhadap *prototype urine analyzer telemetry* yang akan dirancang.

1. Menggunakan rangkaian modul sensor warna TCS230 yang berfungsi sebagai pendeteksi warna urine, TCS230 dapat digunakan untuk mengidentifikasi perbedaan warna yang lebih baik dari LDR atau photodiode.
2. Sistem minimum ATmega328P atau Arduino Uno yang berfungsi sebagai pengendali dan mengolah data sensor warna. Karena tidak membutuhkan terlalu banyak pin I/O seperti ATmega16, ATmega32 atau ATmega128, maka digunakanlah ATmega328 dengan *memory* yang cukup yaitu 2KB RAM dan 32KB ROM, sedangkan pada ATmega8 hanya memiliki 1KB RAM dan 8KB ROM.
3. Menggunakan Arduino Ethernet Shield sebagai penghubung antara Arduino Uno ke jaringan *localhost*. Arduino Ethernet merupakan salah satu protokol yang sering digunakan untuk menghubungkan antara mikrokontroler dengan jaringan komputer menggunakan kabel LAN.
4. SD *card* untuk menyimpan file halaman *web* yang akan dikirim ke PC *client*. SD *card* merupakan *memory external* untuk menyimpan program *interface* laman *web*, agar tidak memakan penggunaan *memory* yang ada pada Arduino.
5. Menggunakan sebuah PC atau notebook yang berfungsi sebagai *interface* berupa tampilan *web* pada *browser*.

6. Menggunakan *power supply* dengan *output* DC 8V dan GND dari *input* 220V. Arduino dapat disupply dengan tegangan *eksternal* melalui *power jack* dengan rentang 7V sampai 12V. Dengan menggunakan IC *regulator* yang mudah ditemukan di toko-toko elektronik, IC 7808 mampu menghasilkan tegangan stabil 8V yang berada pada batas rentang *supply external* Arduino.

C. Blok Diagram Rangkaian

Prototype urine analyzer telemetry ini dirancang pada sebuah *box* dengan *background* putih untuk mendapatkan pembiasan warna urine yang maksimal dari sensor warna, dengan menaruh urine pada wadah dan di dalam ruang tertutup untuk meminimalisir intervensi dari cahaya luar. Sistem ini akan menghasilkan output berupa pembacaan dari pembiasan warna urine, yang kemudian ditampilkan pada *web browser*.



Gambar 9. Blok diagram rangkaian

Dapat dilihat pada Gambar 9 yaitu proses yang dilakukan pada *prototype urine analyzer telemetry*.

1. Sensor warna menggunakan sensor TCS230 untuk mendeteksi perbedaan warna urine yang ditampung pada gelas ukur, urine sebagai media yang akan diidentifikasi oleh sensor.
2. Mikrokontroler yang berfungsi sebagai pusat pengolah data dan pusat pengendali sistem menggunakan Arduino Uno R3.
3. *Power Supply* merupakan sumber tegangan adaptor *power supply* dengan *output* DC 8V dan GND dari *input* 220V atau menggunakan *battery* DC 9V.
4. Arduino Ethernet Shield yang berfungsi sebagai protokol LAN untuk menghubungkan dan mengirim data dari alat ke jaringan komputer
5. SD *card* digunakan untuk menyimpan file halamn *web* yang akan dikirim ke *client*.
6. PC yang berfungsi untuk melihat data pada *web browser*.

D. Perancangan Sistem

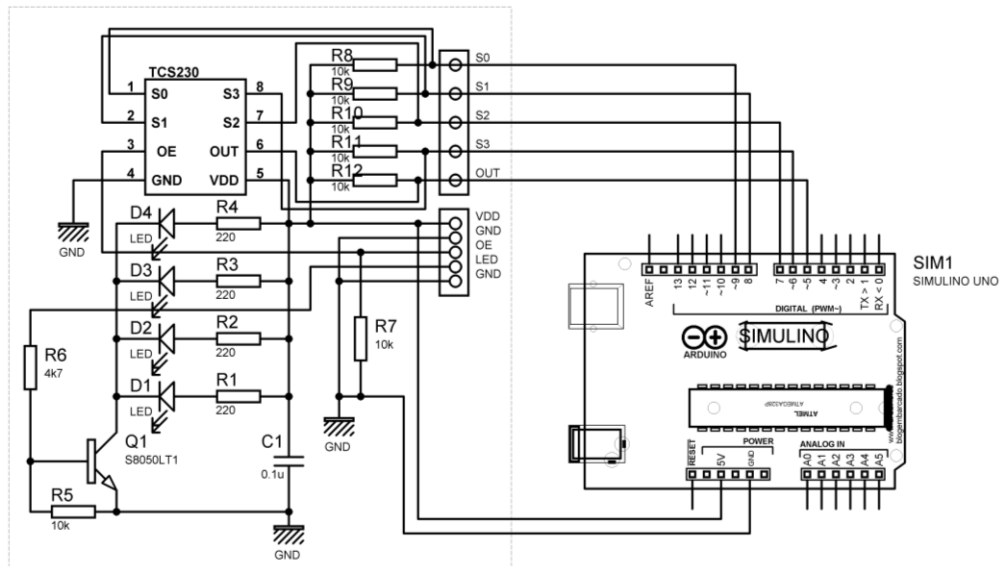
Perancangan sistem *prototype urine analyzer telemetry* ini terbagi dalam 3 bagian, yaitu sensor warna, Arduino Uno, Arduino Ethernet Shield, dan power supply.

1. Sensor warna TCS230

Pendeteksian warna urin dilakukan oleh sensor warna TCS230 dengan menangkap berkas cahaya yang dibiaskan oleh urin sehingga menghasilkan pembiasan berwarna dari led putih pada TCS230 dan diterima pada photodiode sensor TCS230. Pada pin S0 dan S1 sensor

TCS230 digunakan untuk mengatur besar kecilnya frekuensi pada output sensor, sedangkan pin S2 dan S3 sensor TCS230 digunakan untuk menentukan *type* warna cahaya dasar yaitu red, green, dan blue pada *photodiode* sensor TCS230.

2. Arduino UNO Rev 3



Gambar 10. Rangkaian sensor TCS230 yang terhubung ke Arduino

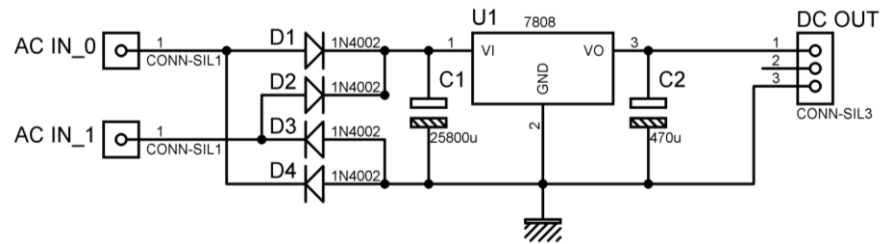
Arduino Uno sebagai pengendali dari *prototype urine analyzer telemetry* dengan mengeksekusi program yang ada di dalamnya. *Port input/output (I/O)* yang digunakan yaitu:

- Port B0 pada ATmega328P, pada Arduino IDE adalah port digital pin 8 terhubung ke S1 sensor TCS230.
- Port B1 pada ATmega328P, pada Arduino IDE merupakan port digital pin 9 terhubung ke S0 sensor TCS230.

- c. Port B2 pada ATmega328P, pada Arduino IDE port digital pin 10 digunakan untuk mengaktifkan/mematikan Ethernet pada komunikasi serial dengan Arduino Ethernet Shield.
 - d. Port D4 pada ATmega328P, pada Arduino IDE yaitu port digital pin 4 digunakan untuk mengatur *chip select* mengaktifkan/mematikan slave SPI pada SD *card* yang ada di Arduino Ethernet Shield.
 - e. Port D5 pada ATmega328P, pada Arduino IDE port digital pin 5 terhubung ke OUT sensor TCS230.
 - f. Port D6 pada ATmega328P, pada Arduino IDE port digital pin 6 terhubung ke S3 sensor TCS230.
 - g. Port D7 pada ATmega328P, pada Arduino IDE port digital pin 7 terhubung ke S2 sensor TCS230.
3. Arduino Ethernet Shield Ver 1

Karena W5100 dan SD *card* menggunakan komunikasi serial sinkron SPI (*Serial Peripheral Interface*), pin 10 untuk mengaktifkan/mematikan Ethernet. Dan menggunakan pin 4 sebagai pin CS/SS untuk mengatur *chip select* mengaktifkan/mematikan *slave* SPI pada SD *card*. Serial sinkron SPI merupakan tipe *protocol* komunikasi serial sinkron dengan 3 jalur, yaitu MISO (*Master In Slave Out*), MOSI (*Master Out Slave In*) dan SCLK (*Serial Clock*). Sehingga port digital pin 10 dan 4 tidak dapat digunakan sebagai I/O untuk sensor atau yang lainnya.

4. Power Supply



Gambar 11. Rangkaian *power supply* regulator 7808

Pada Arduino Uno sumber tegangan *eksternal* melalui *power jack* dapat disupply dengan rentang tegangan sebesar 7V sampai 12V, sehingga untuk *power supply* Arduino Uno dapat digunakan dengan IC regulator 7808 yang menghasilkan tegangan output sekitar 8V. Sedangkan untuk mensupply sensor dan Arduino Ethernet Shield tegangan berasal dari output regulator *internal* yang ada di Arduino Uno. Pada IC regulator tipe 78XX dapat mengeluarkan tegangan stabil jika:

- tegangan masuk lebih tinggi sekurang-kurangnya 2V lebih tinggi dari tegangan yang akan di stabilkannya, karena yang digunakan IC 7808 dengan keluaran 8V maka dalam hal ini tegangan masuk minimal 10V,
- IC 7808 type fisik TO-220AB, hanya sanggup 1A, artinya R beban paling kecil hanya mencapai 8 ohm,

- c. tegangan masuk dapat mencapai maksimal 35V, tetapi power dissipation tidak boleh lebih dari 1,9 watt (tanpa heatsink). Dengan dissipasi daya = $(V_{in} - V_{out}) \times I_{beban}$,
- d. jika menggunakan heatsink yang memadai dan mampu menjaga suhu IC berkisar pada suhu ruang 30°C, maka dissipasi dayanya bisa mencapai 30 watt (Lampiran 8),

Demikian maka pada IC 7808, jika V_{in} adalah 12V dan V_{out} adalah 8V, maka jika tanpa menggunakan heatsink yang memadai.

$$\text{dissipasi daya} = (V_{in} - V_{out}) \times I_{beban}$$

$$1,9 \text{ watt} = (12V - 8V) \times I_{beban}$$

$$I_{beban} = 1,9 \text{ watt} / 4V = 0,475 \text{ Ampere}$$

Dengan beban Arduino Uno, Arduino Ethernet Shield dan sensor TCS230 yang tidak memerlukan arus yang terlalu besar (berkisar 260mA), sehingga dengan transformator 500mA sudah cukup untuk mensupply *prototype urine analyzer telemetry*.

E. Langkah Pembuatan Alat

1. Pembuatan PCB *prototype urine analyzer telemetry*.

Langkah-langkah yang harus dilakukan dalam pembuatan PCB.

- a. Pertama-tama mendesain layout PCB menggunakan software Ares Proteus.

- b. Mencetak gambar layot dengan kertas glossy tinta serbuk.
- c. Selanjutnya mengunting kertas layout PCB yang sudah diprint sesuai dengan ukuran.
- d. Mempersiapkan papan PCB dan mengukur PCB sesuai kebutuhan.
- e. Memotong PCB sesuai dengan ukuran menggunakan alat pemotong atau cutter.
- f. Membersihkan lapisan tembaga PCB dengan stel wool atau amplas sampai bersih. Hingga tidak ada bekas sidik jari atau karat yang menempel pada PCB tersebut.
- g. Menyablon PCB
 - 1) Memasang kertas glossy pada PCB dengan permukaan yang terdapat cetakan gambar menghadap ke sisi PCB polos yang terdapat lapisan tembaganya.
 - 2) Menyiapkan setrika sampai dengan tingkat panas yang sedang. Setrika tidak boleh terlalu panas, karena bisa membuat tembaga pada PCB memuai dan megelembung. Juga tidak boleh terlalu dingin, karena selain akan membuat proses penyablonan lebih lama juga membuat tinta jalur PCB tidak menempel dengan sempurna.
 - 3) Menyetrika kertas glossy dengan penekanan yang sedang dan merata pada setiap bagian PCB.

- 4) Setelah kertas glossy merekat pada PCB, PCB direndam dalam air sampai kertas glossy terangkat dengan sendirinya, atau dengan mengosok kertas dengan perlahan menggunakan tangan mulai dari bagian tengah PCB. Hal ini dilakukan agar tidak merusak tinta yang sudah merekat pada PCB.
- 5) Setelah selesai proses penyablonan, memastikan tidak ada gambar jalur PCB yang terpotong atau mengelupas. Jika ada jalur yang hilang, dapat ditambah atau disambung menggunakan spidol permanen.
- h. Menaburkan FeCl_3 ke dalam nampan nonlogam dan melarutkannya menggunakan air panas. Semakin banyak FeCl_3 akan mempercepat pelarutan. Air yang panas juga akan mempercepat proses pelarutan.
- i. Setelah FeCl_3 larut dalam air, memasukan PCB yang terdapat gambar layout ke dalamnya. Untuk mempercepat proses pelarutan, dapat dilakukan dengan menggoyang-goyangkan nampan secara perlahan. Hal ini dilakukan secara terus menerus sampai semua tembaga yang tidak tertutup tinta di permukaan PCB larut.
- j. Setelah tembaga yang tidak tertutup tinta telah larut, PCB diangkat dan dibersihkan dengan air mengalir.
- k. Mengeringkan PCB dan menggosoknya menggunakan steel wool sampai tinta yang melekat pada jalur PCB bersih. Hal ini dilakukan agar mempermudah penyolderan komponen.

- l. Tahap selanjutnya yaitu *drilling* atau membuat lubang pada PCB. Mata bor yang digunakan memiliki diameter kecil, antara 0,8 dan 1 milimeter untuk komponen dan 3 milimeter untuk mur-baut.
- m. Setelah mengebor PCB, selanjutnya melapisi PCB menggunakan gondorukem pada permukaan tembaga PCB agar tidak mudah teroksidasi, juga untuk mempermudah proses penyolderan. Gondorukem akan membuat timah lebih mudah menempel pada tembaga saat disolder.
- n. Memasang komponen sesuai dengan sekema rangkaian, jangan sampai keliru dalam memasang komponen terutama jika ada polaritas atau kutubnya.
- o. Setelah komponen terpasang dengan benar, dapat merekatkan bagian kaki-kaki komponen dengan cara disolder. Dalam penyolderan harap berhati-hati dan jangan menghirup asap dari timah solder, karena berbahaya bagi kesehatan. Sebaiknya dalam melakukan penyolderan gunakan masker penutup hidung. Selain itu juga jangan terlalu lama menyolder komponen, karena beberapa komponen tidak tahan terhadap panas seperti IC, transistor, LED, bahkan kapasitor bisa kemungkinan akan mati.
- p. Selanjutnya memotong kaki komponen yang telah disolder menggunakan tang potong.
- q. PCB selesai dibuat, kemudian melakukan pengecekan apakah layout yang telah dibuat dapat bekerja dengan baik atau belum.

2. Membuat box prototype urine analyzer telemetry.

Langkah pembuatannya yaitu:

- a. hal yang pertama dilakukan yaitu mengukur kebutuhan box agar semua rangkaian dapat masuk kedalamnya,
- b. menentukan tata letak atau penempatan rangkaian,
- c. langkah selanjutnya adalah mendesain box dengan *software* corel draw,
- d. setelah selesai dengan desain box, *men-cutting akrilic* menggunakan mesin laser,
- e. menyatukannya dengan lem-G tiap-tiap bagian dari *akrilic* yang telah dipotong hingga membentuk box yang telah didesain.
- f. setelah itu, dapat dilakukan penyambungan kabel-kabel instalasi kemudian merakit PCB dan komponen yang telah siap untuk dipasang ke dalam box,
- g. melakukan uji coba pengetesan *prototype urine analyzer telemetry* menggunakan sensor warna untuk meneliti gangguan kesehatan berkemih.

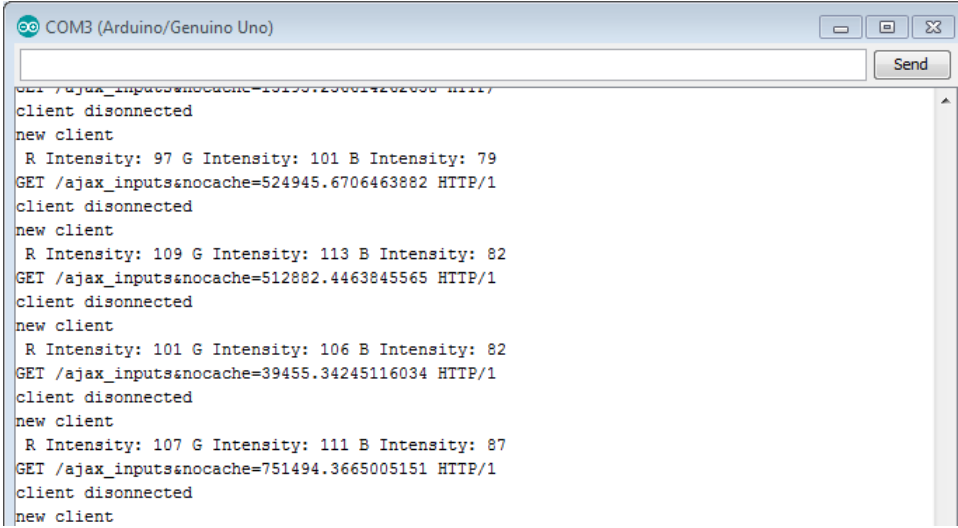
F. Perangkat Lunak

1. Perancangan tampilan

Halaman *web* dibuat sebagai *interface* untuk menampilkan hasil dari pembacaan data sensor warna serta hasil dari analisa warna urine.

Tampilan *web* dapat diakses pada *browser* oleh setiap PC *client* dalam

jaringan *localhost* yang sama dengan mengetikkan alamat IP dari Arduino Ethernet Shield pada URL (*Uniform Resource Locator*) *browser*. Arduino akan menerima permintaan *HTTP request* dari PC *client* dan mengirimkan *HTTP response* sebagai *web server*. Yang di-*request* adalah halaman *web* yang ada pada *SD card*, Arduino akan mengambil halaman *web* yang diminta *client* dan mengirimnya dalam bentuk *response* ke *client*. Pada *web browser client* akan muncul halaman *web* yang diminta. Berikut ini adalah tampilan halaman *web* pada *browser*.

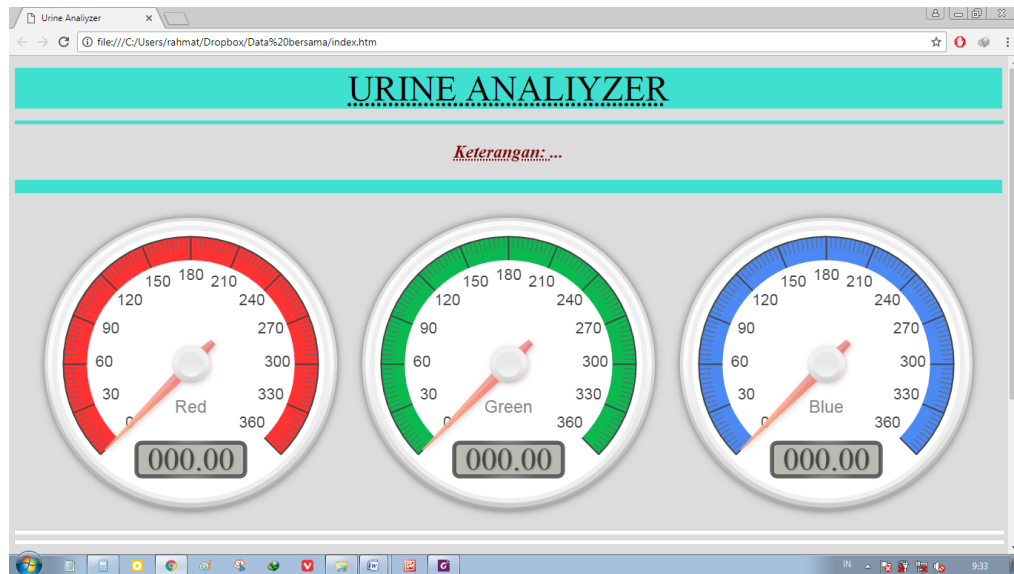


```

COM3 (Arduino/Genuino Uno)
GET /ajax_inputs&nocache=524945.6706463882 HTTP/1.1
client disconnected
new client
R Intensity: 97 G Intensity: 101 B Intensity: 79
GET /ajax_inputs&nocache=524945.6706463882 HTTP/1
client disconnected
new client
R Intensity: 109 G Intensity: 113 B Intensity: 82
GET /ajax_inputs&nocache=512882.4463845565 HTTP/1
client disconnected
new client
R Intensity: 101 G Intensity: 106 B Intensity: 82
GET /ajax_inputs&nocache=39455.34245116034 HTTP/1
client disconnected
new client
R Intensity: 107 G Intensity: 111 B Intensity: 87
GET /ajax_inputs&nocache=751494.3665005151 HTTP/1
client disconnected
new client

```

Gambar 12. Tampilan pada com serial Arduino IDE



Gambar 13. Tampilan *web* pada *PC client*

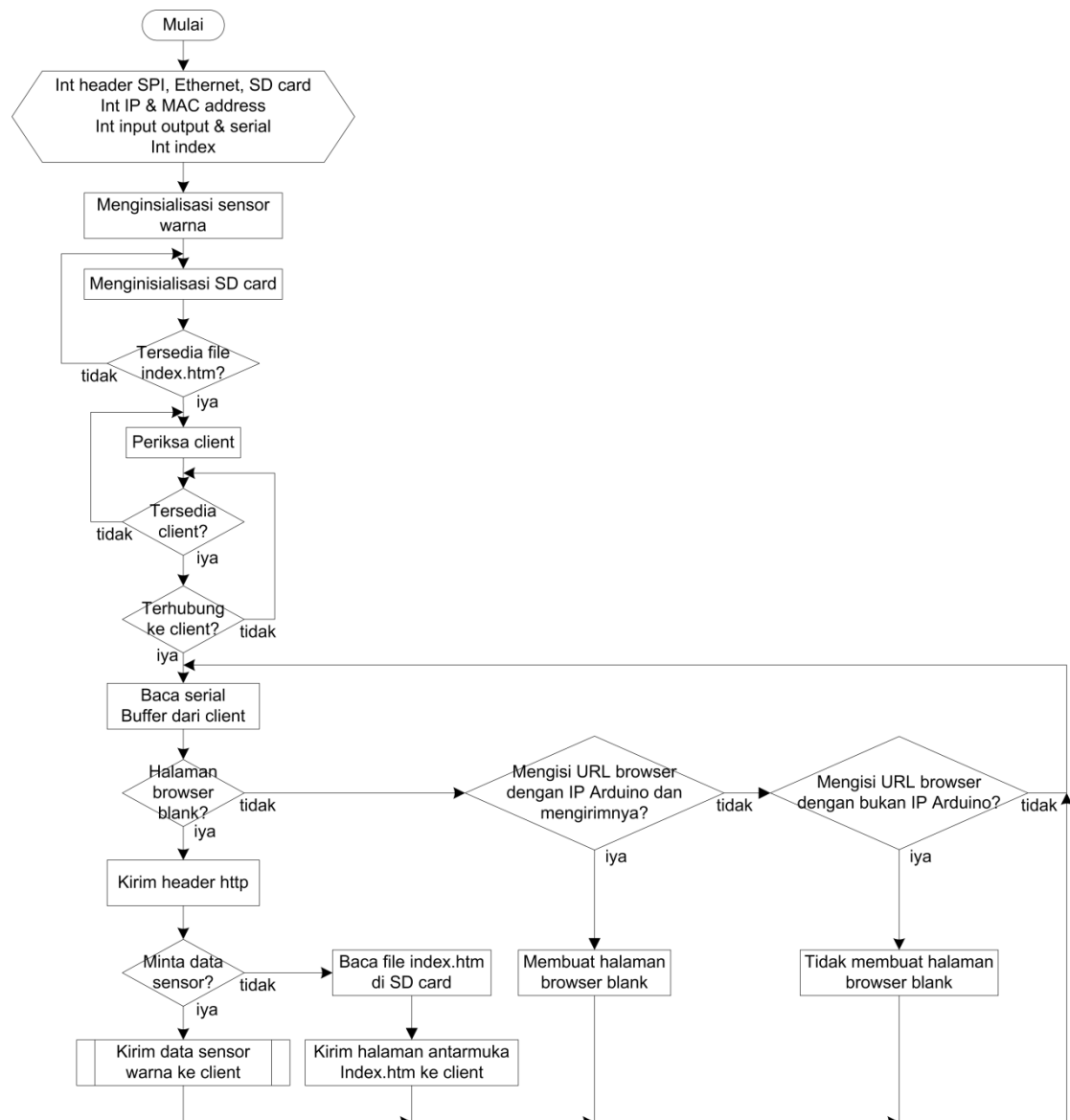
Browser yang ada pada *PC client*, berada pada lapis pertama (lapisan presentasi). Tugas dari lapisan ini yaitu menghadirkan data kepada *client*, menerima *input* dan mengirimkannya dari *client* ke *web server*.

Web server (Arduino Ethernet Shield) berada pada lapis kedua (lapisan menengah). Tugas dari lapisan ini yaitu menerima permintaan data atau proses (*request*) dari *client*, kemudian mengerjakan dan mengembalikan hasilnya (*response*) ke *client*.

Database (Sensor, SD card, dan Arduino Uno) sebagai penyedia data merupakan lapis ketiga yang disebut lapisan data. Tugas dari lapisan ini adalah mengambil data yang dibutuhkan sesuai permintaan *client* dan mengirimkannya kembali serta melakukan proses penambahan dan perbaikan basis data.

2. Perancangan program utama

Gambar 14 menunjukkan *flowchart* program utama. Proses pertama dimulai dengan menginisialisasi port digital Arduino yang terhubung dengan sensor TCS230 dan Arduino Ethernet Shield, seperti komunikasi serial sinkron SPI (*Serial Peripheral Interface*) di Arduino, serta SD card.



Gambar 14. *Flowchart* program utama

Selanjutnya memeriksa keberadaan file index di dalam SD *card* yang nantinya akan ditampilkan pada laman *web browser*, kemudian memeriksa apakah ada permintaan *client* apa tidak. Permintaan *client* dilakukan dengan memasukkan alamat *server* (IP *address* Arduino) pada URL *browser*, jika alamat sesuai maka Arduino Uno akan mengirimkan file index ke *client* lalu di *browser* file tersebut akan diterjemahkan menjadi halaman *web* seperti Gambar 13. Setelah halaman *web* tersedia, *server* akan terus menerus mengirimkan data sensor ke *client*.

3. Subrutin baca data sensor warna

Pada *web server* hasil dari pembacaan sensor berperan seperti *database server* yang diakses oleh *client*, namun *database* tersebut selalu berubah atau di-*update* berdasarkan dari hasil dari pembacaan nilai sensor. Ketika S2 dan S3 dalam kondisi *low* maka *output* merupakan hasil dari pembacaan *filter* warna merah, saat S2 dalam kondisi *low* dan S3 *high* maka *output* merupakan hasil dari pembacaan *filter* warna biru, dan saat S2 dan S3 dalam kondisi *high* maka *output* merupakan hasil dari pembacaan *filter* warna hijau.

```

140 void sensor() { //Baca data sensor warna
141   digitalWrite(s2, LOW);
142   digitalWrite(s3, LOW);
143   red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
144   //red = map (red, 25,72,255,0);
145   digitalWrite(s3, HIGH);
146   blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
147   //blue = map (blue, 25,70,255,0);
148   digitalWrite(s2, HIGH);
149   green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
150   //green = map (green, 30,90,255,0);
151 }

```

Gambar 15. Program *filter* sensor warna pada Arduino IDE

Untuk mengkalibrasi sensor TCS230 dari output berupa arus yang telah terkonversi ke frekuensi RGB dapat dimodelkan ke RGB 255. Untuk mengkalibrasi warna RGB dapat dengan mendeteksi tiga buah warna primer yaitu merah yang sudah diketahui dengan nilai (R=255, G=0, B=0), hijau (R=0, G=255, B=0), dan biru (R=0, G=0, B=255), kemudian nilai intensitas maksimal dan minimal dari warna tersebut digunakan untuk mempersempit range agar hanya terbaca 0-255. Misal pada warna merah primer terbaca (intensitas R=30, G=127, B=92), warna hijau primer (intensitas R=71, G=53, B=87), dan warna biru primer (intensitas R=137, G=109, B=50). Dari ketiga warna tersebut didapatkan intensitas R maksimal adalah 137 dan minimalnya 30, maka dapat dipetakan dengan program:

red = map (red, 30, 137, 255, 0);

begitu pula dengan intensitas G dan B dipetakan menjadi:

green = map (green, 53, 127, 255, 0);

blue = map (blue, 50, 87, 255, 0);

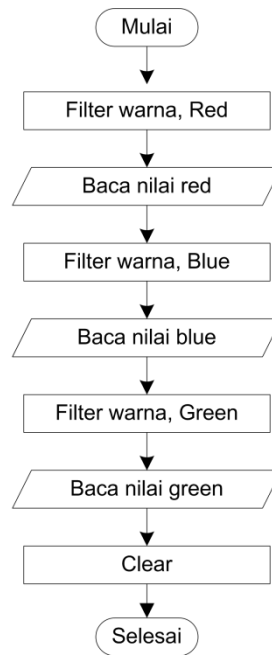
Selain pengkalibrasian menggunakan 3 warna primer RGB, kalibrasi juga dapat menggunakan dua warna yaitu putih dan hitam, dimana warna putih memiliki nilai ($R=255$, $G=255$, $B=255$) sebagai batasan maksimalnya dan hitam ($R=0$, $G=0$, $B=0$) sebagai batasan remaping minimalnya.

Ketika mengkalibrasi menggunakan 2 warna hitam dan putih, perbedaan warna di bedakan berdasarkan intensitas gelap dan terang, sedangkan jika menggunakan 3 warna primer RGB, perbedaan warna di bedakan berdasarkan kepekatan dari kombinasi ketiga warna tersebut (saturasi).

Penentuan scaling (f_o) pada pin S0 dan S1 sensor TCS230 merupakan fungsi penskalaan frekuensi, memungkinkan output sensor dioptimalkan untuk berbagai penghitung frekuensi atau mikrokontroler. Dengan kata lain tergantung pada nilai yang dipilih maka akan menghasilkan sinyal output yang berbeda. Pada scaling 2% nilai output menjadi lebih besar dan berubah-ubah (tidak setabil) sedangkan pada 100% nilai output menjadi terlalu kecil dan selisih dari nilai RGB kecil, sehingga tidak akurat walaupun nilainya setabil. Pada scaling 20% nilai output memiliki range sedang dan cukup setabil.

Tabel 10. Penentuan kondisi TCS230

S0	S1	Output frequency scaling (f_o)		S2	S3	Photodiode type
L	L	Power down		L	L	Red
L	H	2%		L	H	Blue
H	L	20%		H	L	Clear (no filter)
H	H	100%		H	H	Green



Gambar 16. *Flowchart* program subrutin baca data sensor

4. Subrutin kirim data sensor warna ke *client*

Hasil dari pembacaan masing-masing filter warna dikirim dalam bentuk file XML (*eXtensible Markup Language*). File XML membuat data yang ditampilkan dapat dipahami dengan membuat elemen-elemennya sendiri, tidak seperti elemen-elemen HTML yang telah distandarisasi. Data sensor warna dikirim dalam satu paket data dengan mengapitnya menggunakan elemen `<input>` dan `</input>`, setiap data filter warna dipisahkan dengan elemen `<digital>` dan `</digital>`.

Ketika *client* memanggil halaman *web* (`index.htm`) dari *server* (Arduino), *server* merespon panggilan dan mengirimkan halaman *web* yang ada di SD card. Kemudian *client* membuat perintah AJAX (*Asynchronous JavaScript and XML*) ke *server* Arduino dan

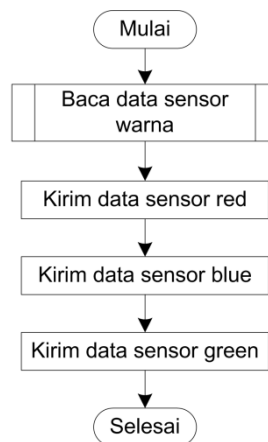
memanggil lebih banyak data (sensor TCS230), *server* mengirimkan data (XML) tersebut dan *client* memutakhirkan laman dengan data tersebut tanpa me-*refresh* halaman *web*. Perintah AJAX berada di dalam file index.htm yang ter-*compile* pada *web browser* saat PC *client* menerima file index.htm dari *server* Arduino.

```

153 void XML_response(EthernetClient cl) {
154   sensor(); //Ambil data sensor warna
155   Serial.print(" R Intensity: "); Serial.print(red, DEC);
156   Serial.print(" G Intensity: "); Serial.print(green, DEC);
157   Serial.print(" B Intensity: "); Serial.print(blue, DEC); Serial.println("");
158
159   cl.print("<?xml version = \"1.0\" ?>");
160   cl.print("<inputs>");
161     cl.print("<digital>"); cl.print(red, DEC); cl.print("</digital>");
162     cl.print("<digital>"); cl.print(green, DEC); cl.print("</digital>");
163     cl.print("<digital>"); cl.print(blue, DEC); cl.print("</digital>");
164   cl.print("</inputs>");
165
166   delay(500);
167 }

```

Gambar 17. Program kirim data ke *client* pada Arduino IDE



Gambar 18. *Flowchart* program subrutin kirim data sensor ke *client*

G. Spesifikasi Alat

Prototype urine analyzer telemetry memiliki spesifikasi sebagai berikut.

1. Bahan pembuatan *prototype urine analyzer telemetry* ini menggunakan bahan akrilik.
2. Pengecekan urine dilakukan dengan penampung botol plastik 50 mili liter sebagai wadah media, dalam hal ini media berupa air urine.
3. Unit masukan yang digunakan adalah sensor warna berupa module TCS230 dengan prinsip sensor menangkap pantulan intensitas cahaya dari media berwarna sehingga sensor mengalami perubahan sinyal listrik yang berupa frekuensi gelombang sebagai data dari filter warna dari media tersebut.
4. Sistem pengendali yang digunakan adalah Arduino Uno Rev 3 dan Arduino Ethernet Shield untuk mengirimkan data ke jaringan komputer.
5. Unit keluaran
 - a. PC untuk menerima data dari jaringan komputer.
 - b. *Web browser* sebagai penampil *interface* data.
6. Tegangan *input* rangkaian yang digunakan adalah DC 8V.

H. Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data penelitian dengan menggunakan dua buah uji, yaitu:

1. Uji fungsional

Pengujian ini dilakukan dengan cara menguji setiap bagian alat berdasarkan karakteristik dan fungsi masing-masing. Pengujian ini dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah bekerja sesuai dengan fungsi dan keinginan.

2. Uji unjuk kerja

Pengujian unjuk kerja alat dilakukan dengan cara melihat unjuk kerja alat. Hal-hal yang perlu diamati antara lain : rangkaian power supply, modul sensor warna, kabel LAN straight, dan hasil tampilan pada *web browser*. Dari pengujian ini akan diketahui kinerja dari alat yang dibuat.

I. Tabel Uji Alat

Tabel 11. Hasil pengukuran *power supply* tegangan tanpa beban

No	Aspek pengukuran	Vin	Vout tanpa beban		Error
			rangkaian	simulasi	
1.	Trafo primer				
2.	Trafo sekunder				
3.	Dioda bridge				
4.	IC regulator				

Tabel 12. Hasil pengukuran *power supply* tegangan dengan beban

No	Aspek pengukuran	Vin	Vout dengan beban		Error
			rangkaian	simulasi	
1.	Trafo primer				
2.	Trafo sekunder				
3.	Dioda bridge				
4.	IC regulator				

Tabel 13. Hasil pembacaan sensor TCS230 dalam model RGB

No	Warna	Jarak dengan sensor	Hasil pembacaan			
			R	G	B	Warna
1.		2,5 cm				
2.		1 cm				
3.		2,5 cm				
4.		1 cm				
5.		2,5 cm				
6.		1 cm				
7.		2,5 cm				
8.		1 cm				
9.		2,5 cm				
10.		1 cm				

Tabel 14. Hasil pengujian tampilan pada *web browser*

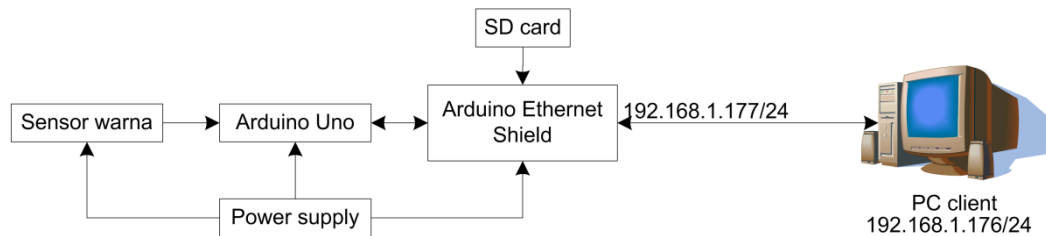
No	Keterangan	Tampilan <i>web browser</i>
1.		
2.		

Tabel 15. Hasil pengujian unjuk kerja pembacaan sampel urine

No	Sampel	Uji (R G B)	Pembacaan	Keterangan
1.	diabetes			
2.	diabetes			
3.	diabetes			
4.	diabetes			
5.	diabetes			
6.	normal			
7.	normal			
8.	normal			
9.	normal			
10.	normal			

J. Pengoprasian Alat

Minimal terdapat sebuah *PC client* sebagai *interface* untuk melihat tampilan *web* pada *browser*.



Gambar 19. Jaringan *localhost* antara *prototype urine analyzer telemetry* dengan satu *PC client*.

Pengoprasian alat ini dapat dilakukan dengan cara sebagai berikut.

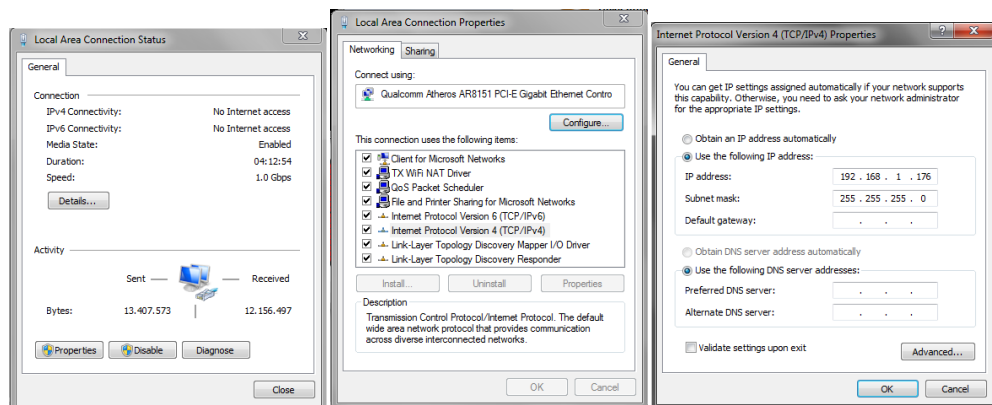
1. Pastikan alat terhubung dengan tegangan AC 220V dan sudah disetabilkan dengan rangkaian adaptor *power supply* maupun *regulator* atau terhubung dengan sumber DC dari *battery*.
2. Hidupkan *prototype urine analyzer telemetry*.
3. Isi gelas ukur atau botol ukur dengan cairan media yang akan di deteksi.
4. Hubungkan *prototype urine analyzer telemetry* dengan kabel LAN tipe *straight* ke *PC client* atau notebook.
5. Atur IP *static* pada *PC client* sehingga berbeda dengan IP pada Arduino Ethernet Shield yang terdapat pada program, karena IP dari Arduino 192.168.1.177/24 maka pada *PC client* bisa kita *setting* IP 192.168.1.176/24

Control Panel >> All Control Panel Items >> Network and Sharing Center >> LAN >> Properties >> Internet Protocol Version 4 (TCP/IPv4)

>> Properties >> IP address = 192.168.1.176

Subnet mask = 255.255.255.0

>> Ok



Gambar 20. Setting IP static pada PC client

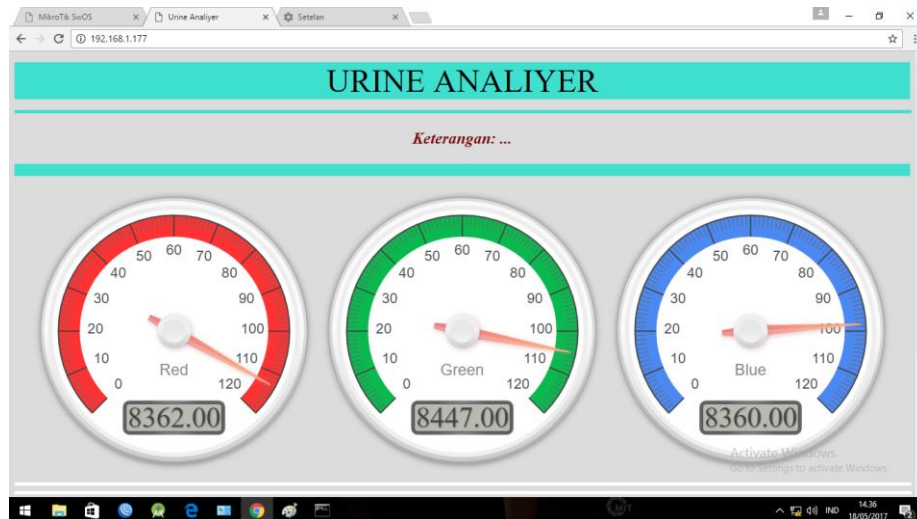
6. Cek koneksi dengan command prompt pada PC client

Start >> command prompt >>

```
C:\Users\>ping 192.168.1.176
ping 192.168.1.177
```

7. Selanjutnya ketikan IP Arduino Ethernet Shield pada browser PC client, maka akan muncul tampilan web server (Arduino sebagai server) yang dikirim dari Arduino ke PC client.

Buka browser >> ketikan 192.168.1.177



Gambar 21. *Interface web dari browser PC client*

BAB IV

PENGUJIAN DAN PEMBAHASAN

A. Pengujian

Pengujian fungsional *prototype urine analyzer telemetry* dilakukan dengan cara bertahap. Pengujian terdiri dari pengujian tegangan *power supply*, pembacaan sensor, dan tampilan pada *web browser*. Serta pengujian unjuk kerja, uji unjuk kerja dilakukan dengan mengoprasikan keseluruhan alat yang telah dibuat sesuai dengan tujuannya. Setelah alat diuji perbagian secara fungsional, kemudian diuji secara keseluruhan dengan memberikan sampel media zat cair berupa urine atau air berwarna.

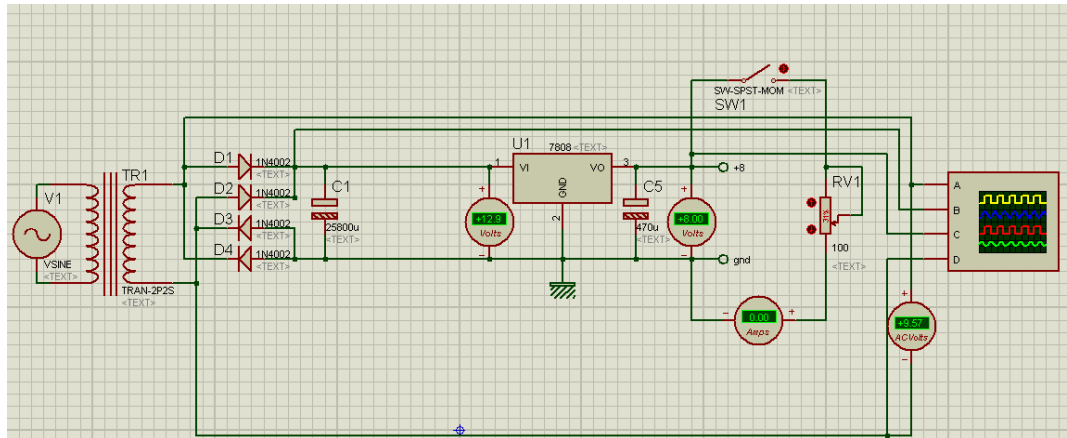
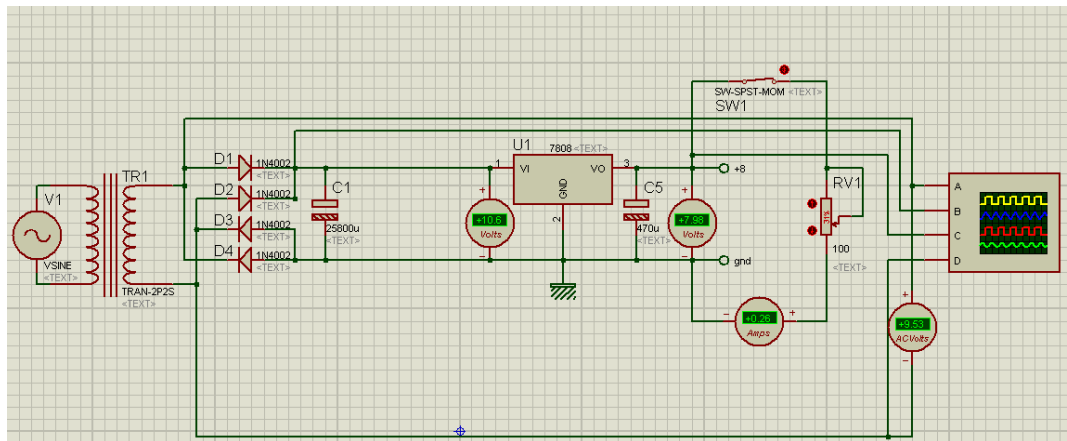
1. Pengujian power supply

Pengujian *power supply* dilakukan dengan mengukur V_{in} dan V_{out} pada trafo, dioda penyearah, IC regulator, dan input V_{in} Arduino. Pada trafo 500mA ketika tanpa beban pengukuran trafo pada *primer* dan *sekunder* menunjukkan hasil pengukuran sebesar 235V dan 9,5V (AC).

Pada Tabel 17, pengukuran dilakukan dengan beban berupa sensor, Arduino Uno, dan Arduino Ethernet Shield dengan pembacaan Ibeban sebesar 260mA (arus *output power supply*). Dari hasil pengukuran pada rangkaian dan simulasi Proteus didapatkan selisih yang tidak terlalu jauh.

Tabel 16. Hasil pengukuran *power supply* tegangan tanpa beban

No	Aspek pengukuran	Vin	Vout tanpa beban		Error
			rangkaian	simulasi	
1.	Trafo primer	235 VAC	235 VAC	235 VAC	0%
2.	Trafo sekunder	235 VAC	9,5 VAC	9,57 VAC	0,73%
3.	Dioda bridge	9,5 VAC	12,5 VDC	12,9 VDC	3,10%
4.	IC regulator	12,5 VDC	8,3 VDC	8 VDC	3,75%

Gambar 22. Rangkaian *power supply* tanpa beban pada simulasi ProteusGambar 23. Rangkaian *power supply* dengan beban pada simulasi Proteus

Pada simulasi Proteus diberi beban R dengan potensio untuk mendapatkan arus sebesar 260mA dan didapat nilai 31% dari 100Ω yaitu 31Ω, dengan kata lain beban sensor, Arduino Uno, dan Arduino Ethernet Shield yaitu sebesar 31Ω.

Sedang jika dihitung dari hasil pengukuran rangkaian yaitu:

$$R = \frac{V}{I}$$

$$R = \frac{8,1V}{0,26A} = 31,15\Omega$$

Dengan beban 31,15Ω yang berarti tidak melanggar syarat minimum IC 7808 yang hanya mampu diberi beban minimal 8Ω (untuk kemampuan maksimal 1A tanpa heatsink yang memadai). Serta dengan selisih Vout tanpa beban pada *power supply* antara hasil simulasi dan pada rangkaian yaitu $8,3V - 8V = 0,3V$ (DC).

$$Error = \frac{\text{selisih pengukuran}}{\text{nilai pembanding}} \times 100\%$$

$$Error = \frac{0,3V}{8V} \times 100\% = 3,75\%$$

Maka didapatkan *error output* tegangan *power supply* tanpa beban sebesar 3,75%

Tabel 17. Hasil pengukuran *power supply* tegangan dengan beban

No	Aspek pengukuran	Vin	Vout dengan beban		Error
			rangkaian	simulasi	
1.	Trafo primer	235 VAC	235 VAC	235 VAC	0 %
2.	Trafo sekunder	235 VAC	9 VAC	9,53 VAC	5,56%
3.	Dioda bridge	9,5 VAC	9,7 VDC	10,6 VDC	8,49%
4.	IC regulator	12,5 VDC	8,1 VDC	7,98 VDC	1,50%

Sedangkan selisih V_{out} dengan beban pada *power supply* antara hasil simulasi dan pada rangkaian yang telah dibuat yaitu $8,1V - 7,98V = 0,12V$ (DC).

$$Error = \frac{\text{selisih pengukuran}}{\text{nilai pembandingan}} \times 100\%$$

$$Error = \frac{0,12V}{7,98V} \times 100\% = 1,5\%$$

Maka didapatkan *error output* tegangan *power supply* dengan beban sebesar 1,5%

2. Pengujian pembacaan sensor warna TCS230

Pengujian pembacaan sensor TCS230 dilakukan dengan membandingkan nilai pembacaan dengan sampel uji media berwarna, dari hasil nilai RGB yang terbaca kemudian dikonversi ke warna dan dapat diketahui perbedaan tingkat keabu-abuan (*value*) maupun kecerahan warna (*chroma*) yang mampu terbaca.

Tabel 18. Hasil pembacaan sensor TCS230 dalam model RGB

No	Warna	Jarak dengan sensor	Hasil pembacaan			
			R	G	B	Warna
1.	Biru tua	2,5 cm	143	156	195	
2.	Biru tua	1 cm	36	35	110	
3.	Hijau muda	2,5 cm	160	193	164	
4.	Hijau muda	1 cm	72	135	-188	
5.	Merah	2,5 cm	186	169	177	
6.	Merah	1 cm	184	-82	-121	
7.	Hijau tua	2,5 cm	153	204	164	
8.	Hijau tua	1 cm	-14	80	-151	
9.	Biru muda	2,5 cm	148	190	201	
10.	Biru muda	1 cm	-11	149	195	

Pengujian sampel dari jarak 1 cm memiliki hasil warna yang lebih cerah daripada ketika sampel diletakan diatas sekat akrilik dengan jarak 2,5 cm dari sensor. Hal ini menunjukkan ketika semakin jauh media pembacaan maka frekuensi cahaya yang mampu ditangkap sensor semakin pudar, namun dalam jarak 2,5 cm sensor masih mampu menangkap perbedaan warna cahaya yang dipantulkan dari warna sampel.





Gambar 24. Sampel warna pengujian sensor

Background box berwarna putih akan memantulkan semua frekuensi warna dari media sampel, sehingga sensor akan lebih banyak menangkap pantulan warna media sampel dari pada dengan *background* warna lainnya.

3. Pengujian tampilan *web browser*

Pengujian tampilan web dilakukan dengan cara, apakah data yang terbaca dari sensor dapat ditampilkan pada web browser. Dan didapatkan hasil bahwa file index dapat terkirim ke *client* dan diterjemahkan menjadi laman *web* seperti pada Tabel 19 dibawah ini.

Tabel 19. Hasil pengujian tampilan pada *web browser*




No	Keterangan	Tampilan <i>web browser</i>
1.	Urine warna kuning bening	
2.	Urine warna kuning muda dan keruh	






4. Pengujian unjuk kerja



Pengujian unjuk kerja, pada tahap ini pengujian dilakukan menggunakan sampel berupa urine sebanyak 100ml yang diletakan didalam sebuah botol plastik trasnparant, kemudan hasil dari pembacaan sensor diklasifikasikan sesuai dengan data base yang telah dibuat pada program. Pada tahap uji keseluruhan, sampel yang telah diuji coba *prototype urine analyzer telemetry* mampu digunakan untuk mengidentifikasi perbedaan warna walaupun tidak memiliki tingkat akurasi yang tinggi.

Pengujian unjuk kerja dilakukan dengan menggunakan 10 sampel (warna urine) dari 2 kategor berbeda dengan sepesifikasi warna yang telah ditentukan. Dari 10 sampel yang diuji, sampel no 1 sampai 5 merupakan sampel dari urine yang telah didiagnosa, ditetapkan atau memiliki riwayat sebagai sakit diabetes, sedangkan sampel no 6 sampai 10 merupakan sampel urine orang normal.

Tabel 20. Hasil pengujian unjuk kerja pembacaan sampel urine

No	Sampel	Uji (R G B)	Pembacaan	Keterangan
1.	 diabetes	205 242 243	diabetes	sesuai
2.	 diabetes	208 261 243	diabetes	sesuai
3.	 diabetes	208 252 261	diabetes	sesuai

4.	 diabetes	222 261 249	diabetes	sesuai
5.	 diabetes	231 280 309	normal	kesalahan pembacaan
6.	 normal	227 265 285	normal	sesuai
7.	 normal	222 258 273	normal	sesuai
8.	 normal	231 279 309	normal	sesuai

9.	 normal	233 279 307	normal	sesuai
10.	 normal	229 268 285	normal	sesuai

Berdasarkan 10 sampling yang telah di ujikan, terdapat satu kesalahan pembacaan pada sampel no5. Maka didapatkan rata-rata error sebesar :

$$\begin{aligned} \text{Rata-rata } error &= \frac{\text{jumlah } error}{\text{jumlah data pengujian}} \times 100\% \\ &= \frac{1}{10} \times 100\% = 10\% \end{aligned}$$

Kesalahan pembacaan dapat dikarenakan oleh beberapa faktor, seperti penempatan posisi sampel urine yang tidak tepat sehingga menghasilkan sudut pembiasan cahaya yang berbeda, kurangnya data base pengidentifikasian serta pengaruh dari cahaya luar yang menginterfensi pembacaan sensor. Kesetabilan tegangan input juga mampu membuat hasil pembacaan menjadi tidak setabil.

B. Pembahasan

Berdasarkan hasil pengujian secara keseluruhan, serta hasil dari perancangan software dan hardware yang telah dibuat. Maka dapat diuraikan beberapa permasalahan yang berhubungan dengan kinerja alat dalam pembahasan ini, yaitu:

1. *Power supply*

Dari pengujian tegangan *power supply* dan pengujian pembacaan sensor, pada rangkaian catu daya tanpa beban memiliki presentase *output error* sebesar 3,75% dan presentase *output error* dengan beban sebesar 1,5%. *Power supply* tersebut digunakan untuk mencatu Arduino Uno yang membutuhkan tegangan *input* 7-12 V, dengan kebutuhan arus total rangkaian yang berkisar 260mA maka dengan menggunakan sebuah trafo NON-CT 0,5 A untuk mensupply keseluruhan rangkaian dirasa sudah cukup. IC regulator 7808 dengan disupply dari dioda penyearah sebesar 12,5 V (hasil pengukuran) maka:

$$\text{dissipasi daya} = (V_{in} - V_{out}) \times I_{\text{beban}}$$

$$1,9 \text{ watt} = (12,5V - 8V) \times I_{\text{beban}}$$

$$I_{\text{beban}} = 1,9 \text{ watt} / 4,5V = 0,422 \text{ Ampere}$$

Dengan arus beban maksimal yang bisa dilewatkan rangkaian regulator sebesar 422mA sudah mencukupi kebutuhan rangkaian yang hanya 260mA, dan tidak diperlukan penggunaan trafo melebihi 0,5 A. pada sensor TCS230 dan Arduino Ethernet Shield tegangan *supply* didapat

dari regulator *internal* yang ada di Arduino, memberikan tegangan sebesar 5 V.

2. Mikrokontroler

Mikrokontroler Arduino Uno menjadi pusat kendali untuk membaca data dari sensor TCS230. Selain berperan untuk mengendalikan proses pembacaan sensor warna dengan 4 filter (Red, Blue, Clear, dan Green), juga sebagai pengendali untuk memanggil file *web* (Lampiran 6.) yang akan dikirimkan ke *browser* saat *client* meminta file dengan memasukan alamat IP Arduino Ethernet Shield pada URL *browser client*, dalam program ini IP disetting sebagai IP *static* yaitu dengan alamat 192.168.1.177 dan subnet mask 255.255.255.0

Ketika *client* sudah mengakses file *web* yang dikirimkan oleh Arduino (index.htm), Arduino akan terus menerus mengirimkan data sensor ke *client* tanpa perlu *refresh* laman *web browser*.

Pada Arduino Ethernet Shield terdapat *slot SD card* yang digunakan untuk memasukan memori *micro SD card* sebagai media penyimpanan file *web* (index.htm). *SD card* tersebut menggunakan komunikasi serial sinkron (*Serial Peripheral Interface*) yang merupakan tipe protokol komunikasi serial dengan 3 jalur, yaitu MISO (*Master In Slave Out*), MOSI (*Master Out Slave In*) dan *SCLK* (*Serial Clock*) dengan mengatur port digital pin 4 pada Arduino untuk mengendalikan *chip select* mengaktifkan atau mematikan slave SPI pada *SD card* tersebut.

Berikut merupakan potongan program untuk menginisialisasi SD *card*, yaitu:

```
Serial.println("Inisialisasi SD card...");
if (!SD.begin(4)) {
    Serial.println("ERROR - Inisialisasi SD card !!!");
    return;
}
```

Jika SD *card* belum di masukan ke dalam *slot* SD *card* di Arduino atau SD *card* tidak terbaca maka akan di inisialisasi sebagai *error*, dan akan terus mengulang program ini sampai SD *card* terdeteksi.

```
pinMode(10, OUTPUT);
digitalWrite(10, HIGH);
```

Dan program diatas untuk mengaktifkan atau mematikan Ethernet pada port digital pin 10 Arduino.

3. Sensor

Hasil dari keluaran sensor TCS230 berupa perubahan sinyal listrik yang berupa frekuensi gelombang sebagai data dari *filter* warna dari media sampel dan ditampilkan sebagai nilai RGB. Pada alat ini sensor TCS230 dihubungkan dengan port digital, yaitu:

- a. Pin 9 Arduino yang terhubung ke S0 sensor TCS230
- b. Pin 8 Arduino terhubung ke S1 sensor TCS230
- c. Pin 7 ke S2 sensor TCS230
- d. Pin 6 terhubung ke S3 sensor TCS230, dan
- e. Pin 5 terhubung ke OUT sensor TCS230

Pin S0 dan S1 pada sensor warna TCS230 digunakan untuk mengatur besar kecilnya dari *output* frekuensi *scaling*. Diantara 4 pengaturan, pada program ini digunakan *scaling* 20% yaitu dengan S0 disetting

dalam kondisi *high* dan S1 pada kondisi *low* (Tabel 10.), ketika menggunakan *scaling* 2% data *output* memiliki range yang terlalu besar dan outputnya tidak stabil karena menunjukkan data yang berubah-ubah sedangkan ketika menggunakan *scaling* 100%, data *output* memiliki range atau batasan yang kecil namun tidak begitu sensitif karena memiliki nilai perubahan yang tipis atau hampir sama ketika mendeteksi warna yang berbeda.

Dengan mensetting *scaling* 20% dan memberi batasan baru menggunakan remapping, maka dapat dihasilkan data *output* baru yang lebih tepat. Sedangkan penggunaan pin S2 dan S3 pada sensor TCS230 difungsikan untuk mengatur *filter* yang dipilih saat membaca perbedaan warna. Data yang dihasilkan memiliki hasil yang stabil dan cukup sensitif ketika digunakan untuk membaca 3 warna primer cahaya (Red, Green, Blue), hal ini dikarenakan sesuai dengan data sheet yang hanya memiliki 4 *filter* (Red, Blue, Clear, dan Green) sehingga memiliki sensitivitas pada 3 warna primer tersebut. Dengan begitu untuk mendapatkan persepsi warna yang sesuai, maka hasil pembacaan dari *scaling* 20% dikalibrasi dengan 3 warna primer (Red, Green, Blue) dengan warna yang telah ditentukan berdasarkan model RGB, yaitu: warna primer merah dengan nilai model RGB (Red=255, Green=0, dan Blue=0), kemudian hasil dari pembacaan sensor TCS230 dengan *scaling* 20% tersebut digunakan sebagai pengkalibrasi warna dengan menjadikannya batasan maksimal dan minimal dari nilai *output*

warna merah. Begitu pula dengan warna primer yang lain, yaitu: warna primer hijau dengan nilai model RGB (R=0, G=255, dan B=0), serta warna primer biru dengan nilai model RGB (R=0, G=0, dan B=255), dari ketiga warna primer tersebut dapat dijadikan acuan untuk remaping atau mempersempit range pembacaan warna tersebut.

Pada uji trials and error (Lampiran 12) digunakan sebagai acuan pembandingan warna sampel yang akan diidentifikasi, atau sebagai data base (program index.htm) dari sampel yang ingin kita deteksi. Selanjutnya Tabel 18 merupakan tahap pengujian sensor TCS230 setelah ditetapkan warna sampel yang digunakan sebagai acuan pengidentifikasian.

Saat *client* sudah memanggil file halaman *web* (index.htm) dengan memasukkan alamat IP Arduino ke URL *browser*, maka halaman *web* akan dikirim ke *client* dan membuat perintah untuk meminta data sensor dengan fungsi AJAX (*Asynchronous JavaScript and XML*), yang akan mengupdate hasil pembacaan sensor tanpa perlu *merefresh* laman *web* tersebut. Berikut ini potongan program AJAX pada file index yang disimpan di SD card:

```
request.open("GET", "ajax_inputs" + nocache, true);
request.send(null);
setTimeout('GetArduinoInputs()', 200);
```

AJAX (*Asynchronous JavaScript and XML*) merupakan sebuah sistem untuk mengirim dan menerima data dari *server* tanpa penyegaran laman (*page refresh*). Mengambil data dengan metode GET, maka kita

harus mengirimkan sesuatu ke *server*, tapi karena kita tidak mengirimkan apa-apa maka kita kirimkan nilai kosong ke *server*.

Pengidentifikasian data sensor TCS230 dilakukan pada *web browser* atau disebut *client side* (pengolahan data *web server* pada *client*), sehingga Arduino sebagai *server* hanya berperan untuk memberikan data ke *client*.

```
var request = new XMLHttpRequest();
request.onreadystatechange = function()
```

Fungsi XMLHttpRequest, dimana suatu halaman *web* dapat direquest dari *server* dan diterima hasilnya tanpa perlu terjadi *refresh* pada halaman *web*. Sedangkan fungsi yang ada dibawahnya (request.onreadystatechange), merupakan sebuah fungsi dimana nanti kita akan memperoleh status dari request file yang telah dilakukan.

Untuk mengambil data pertama yang diapit elemen digital pada responseXML, kemudian menyimpannya sebagai variabel red. Dan untuk mengambil data kedua sebagai variabel green serta data ketiga sebagai variabel blue maka digunakan fungsi berikut:

```
red = this.responseXML.getElementsByTagName('digital')
[0].childNodes[0].nodeValue;
green = this.responseXML.getElementsByTagName('digital')
[1].childNodes[0].nodeValue;
blue = this.responseXML.getElementsByTagName('digital')
[2].childNodes[0].nodeValue;
```

Program pada file index yang digunakan untuk mengambil data digital tersebut akan membuat *web browser* menerima data dari Arduino. Data sensor warna yang dikirim dari Arduino, dikirim dalam satu paket data dengan mengapitnya menggunakan elemen <input> dan </input>, setiap

data filter warna dipisahkan dengan elemen `<digital>` dan `</digital>`.

(Gambar 17.)

Setelah data diterima pada *web browser*, nilai-nilai tersebut diidentifikasi atau dikelompokkan menggunakan fungsi (if else) pada file *web* (index.htm) dan kemudian ditampilkan pada *browser client*.

Berikut potongan program untuk mengelompokkan data warna pada file *web*:

```
if (red < 276 && red > 272 && green < 356 && green > 352
&& blue < 384 && blue > 380) {
document.getElementById("ambil_digital").innerHTML =
"normal" ; }
else if (red > 237 && red < 241 && green > 297 && green <
301 && blue > 313 && blue < 317) {
document.getElementById("ambil_digital").innerHTML =

"normal" ;
```

Dengan menggunakan sensor TCS230 kita dapat mengetahui warna zat cair maupun benda padat. Berdasarkan *website* <http://mediskus.com> dan <http://www.eyuana.com> urine memiliki berbagai macam warna yang dapat mengindikasikan beberapa penyakit salah satunya adalah diabetes melitus. Ada banyak penyebab yang dapat membuat warna urine kita tidak seperti biasanya, diantaranya ada penyebab tidak berbahaya dan berlangsung sementara seperti makan makanan tertentu atau minum beberapa jenis obat. Sedangkan ada penyebab lainnya mungkin lebih serius dan warna urine yang tidak seperti biasanya itu dapat menunjukkan adanya kondisi medis atau penyakit tertentu.

Dengan sensor TCS230 kita bisa mengidentifikasi warna urine untuk mengetahui penyakit. Pada pembacaan sensor warna, sensor sangat

dipengaruhi dari jarak antara media warna yang akan dideteksi dengan sensor. Selain itu input kesetabilan sumber tegangan yang digunakan untuk mensupply juga sangat berpengaruh pada kesetabilan pembacaan sensor. Pada pembacaan sampel media zat cair secara berturut-turut didapatkan hasil yang cukup setabil pada warna primer. Hal ini karena sistem pengkalibrasian dari sensor TCS230 menggunakan warna *additive color system* sebagai dasar filter pada sensor TCS230 sesuai datasheet.

BAB V

SIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil percobaan yang telah dilakukan pada *prototype urine analyzer telemetry* menggunakan sensor warna untuk mendeteksi penyakit diabetes pada seseorang dapat disimpulkan sebagai berikut:

1. Perangkat keras *prototype urine analyzer telemetry* telah berhasil dibuat dengan menggunakan beberapa komponen dan rangkaian, diantaranya: rangkaian regulator sebagai penstabil tegangan pada *power supply*, rangkaian sensor TCS230 sebagai *input*, Arduino UNO untuk pengendali dan pengolah data, dan Arduino Ethernet Shield sebagai menghubungkan antara mikrokontroler dengan jaringan komputer melalui kabel LAN untuk ditampilkan pada browser PC.
2. Perangkat lunak yang diaplikasikan dalam sistem ini menggunakan pemrograman bahasa Arduino atau bahasa C yang telah disederankan pada Arduino IDE. Sedangkan untuk interface laman web, menggunakan bahasa htm atau html. Berdasarkan pengujian yang telah dilakukan, perangkat lunak ini dapat berjalan dengan baik untuk membaca data input sensor, mengirim data ke *client* atau ke *browser* PC dan menampilkannya.

3. Unjuk kerja *prototype urine analyzer telemetry* menggunakan sensor warna untuk mendeteksi penyakit diabetes pada seseorang secara keseluruhan telah berjalan dengan baik yaitu saat sensor menangkap perubahan rona warna, tingkat terang-gelap, dan intensitas warna cahaya yang dipantulkan dari permukaan media sampel, hasilnya akan diolah di Arduino kemudian untuk dikirimkan ke *client* PC. Warna cahaya yang ditangkap diolah dengan model RGB dan di remaping untuk mempersempit *range* hasil pembacaan sensor.

B. Keterbatasan Alat

Prototype urine analyzer telemetry yang telah dibuat memiliki keterbatasan, antara lain:

1. Pada alat ini hanya mampu menunjukkan kategori beberapa penyakit yang dibedakan berdasarkan warna sampel urine.
2. Data yang ditampilkan hanya sekilas, karena tidak adanya data logging untuk menyimpan data pembacaan sensor ke *SD card*.
3. Hasil pembacaan sensor TCS230 lebih peka terhadap warna benda solid dari pada benda cair yang tembus cahaya.
4. Pembacaan sensor TCS230 mudah terintervensi oleh cahaya, sehingga posisi dan jarak mempengaruhi hasil pantulan cahaya media

C. Saran

Pembuatan proyek akhir ini memiliki beberapa kekurangan, sehingga diperlukan pengembangan lebih lanjut. Saran yang membangun diperlukan untuk menyempurnakan proyek akhir ini, antara lain sebagai berikut:

1. Untuk mendeteksi urine sebaiknya digunakan berbagai macam parameter, agar dapat lebih akurat. Dikarenakan jika hanya menggunakan sensor warna TCS230 data yang didapat untuk membaca warna zat cair kurang sensitif. Mungkin dapat dikembangkan menggunakan kamera untuk mendapatkan hasil yang lebih baik.
2. Sistem ini masih berupa *prototype*, sehingga perlu penelitian lebih lanjut untuk menguji keefektifan alat agar dapat diterapkan dalam kehidupan nyata.
3. Pemeriksaan urine disarankan hanya dijadikan sebagai tes awal saja, perlu pemeriksaan lainnya agar diagnosa penyakit menjadi lebih tepat.
4. Penggunaan warna *background box* sebaiknya digunakan warna putih, agar tidak mempengaruhi hasil pembacaan sensor.

DAFTAR PUSTAKA

- Abidin, M. S. (2015). Desain Sistem Pengukuran Gula Darah Menggunakan Metode Non-Invasive Berbasis Spektrofotometer Sederhana. *Jurnal Teknik Elektromedik Stikes Mandala Waluya Kendari*.
- Alimul, H. A. A. (2006). *Pengantar Kebutuhan Dasar Manusia: Aplikasi Konsep dan Proses Keperawatan Buku 2*. Jakarta: Salemba Medika
- American Diabetes Association. (2010). Diagnosis and Classification of Diabetes Mellitus. *Diabetes Care* Vol. 33, p 62-69
- Amir. (2015). Kadar Glukosa Darah Sewaktu Pada Pasien Diabetes Melitus Tipe 2 di Puskesmas Batu Kota Manado. *Jurnal e-Biomedik (eBm)*, Vol. 3, No. 1.
- Anna, L.K. (2011). *Alzheimer Penyakit Diabetes Otak?*. Diakses pada tanggal 25 Januari 2018, dari <https://lifestyle.kompas.com/read/2016/08/15/132200823/Alzheimer.Penyakit.Diabetes.Otak>.
- Arduino e Cia. (2014). Sensor de reconhecimento de cor TCS230 / TCS3200. Diakses pada tanggal 26 juli 2017, dari <https://www.arduinoecia.com.br/2014/02/sensor-de-reconhecimento-de-cor-tcs230.html>
- Aslamia, S. (2015). Robot Pendeteksi Manusia Sebagai Sistem Keamanan Ruangan Menggunakan Sensor PIR Dengan Media Komunikasi XBEE Berbasis Arduino Leonardo. Tesis, tidak dipublikasikan. *Politeknik Negeri Sriwijaya, Palembang*.
- Candogan, Mike. (2016). *Urine Colour*. Diakses pada tanggal 26 juli 2017, dari <https://lifeinthefastlane.com/investigations/urine-colour/>
- Candogan, Mike. (2010). *Urinalysis_2*. Diakses pada tanggal 5 november 2016, dari https://lifeinthefastlane.com/wp-content/uploads/2010/04/Urinalysis_2.jpg
- Daniakso, F. R. (2016). LKP : Rancang Bangun Aplikasi Pengaduan Keluhan Pelanggan Berbasis Web PHP Pada PDAM Surya Sembada Surabaya. Undergraduate thesis, tidak dipublikasikan. *Institut Bisnis dan Informatika Stikom Surabaya*.
- Dickson Kho. (2017). *Jenis-jenis IC Voltage Regulator (Pengatur Tegangan)*. Diakses pada tanggal 27 juli 2017, dari

<http://teknikelektronika.com/jenis-ic-voltage-regulator-pengatur-tegangan/>

- Dickson Kho. (2017). *Rangkaian-IC-Fixed-Voltage-Regulator*. Diakses pada tanggal 27 juli 2017, dari <http://teknikelektronika.com/wp-content/uploads/2014/10/Rangkaian-IC-Fixed-Voltage-Regulator.jpg?x22079>
- Dorland, W. A. N. (2010). *Dorland's Illustrated Medical Dictionary (29 ed.d)*. Hartanto.H. et al. (alih bahasa). Jakarta: EGC.
- Elektronika Dasar. (2012). *Regulator Tegangan Positif 78XX*. Diakses pada tanggal 28 Maret 2018, dari <http://elektronika-dasar.web.id/regulator-tegangan-positif-78xx/>
- Elektronika Dasar. (2012). *Sensor Warna TCS230*. Diakses pada tanggal 2 Januari 2017, dari <http://elektronika-dasar.web.id>.
- Fatimah, R. N. (2015). Diabetes Melitus Tipe 2. *Fakultas Kedokteran Universitas Lampung Lampung*, 4(artikel review), 93`C101.
- Hasnah. (2009). Pencegahan Penyakit Diabetes Mellitus Tipe 2. Media Gizi Pangan, Vol. VII, Edisi 1, Januari - Juni 2009. dari <http://jurnalmediagizipangan.files.wordpress.com//2012/03/01/1-pencegahan-penyakit-diabetes-melitus.pdf>
- Ineke, V. (2017). Perbedaan Hasil Protein Urin Metode Carik Celup dan Metode Rebus Asam Asetat 6%. Diploma thesis, tidak dipublikasikan. *Muhammadiyah University of Semarang*.
- John, S. (2011). The Pursuit of Non-Invasive Glucose: Huting the Deceitful, Turkey
- Kassim, N. M., Kit, S. Y. H. (2013). *Non Invasive Blood Glucose Measurement Using Temperature-Based Approach*. Diakses pada tanggal 9 Maret 2018, dari <https://id.scribd.com/document/253278592/Non-Invasive-Blood-Glucose-Measurement-Using-Temperature-Based>.
- Palensyah, P. (2014). Aplikasi Sensor Warna TCS230 Pada Sistem Kendali Robot Line Follower. Other Tesis, tidak dipublikasikan. *Politeknik Negeri Sriwijaya*.
- Perkeni. (2011). *Konsensus Pengelolaan dan Pencegahan Diabetes Melitus tipe 2 di Indonesia 2011*. Jakarta: Perkumpulan Endokrinologi Indonesia.

- Rokim, Achmad. (2015). Rancang Bangun Alat Deteksi Dehidrasi Menggunakan LED dan Fotodioda Melalui Warna Urine. Skripsi, tidak dipublikasikan. *Program Studi Fisika Universitas Islam Negeri Sunan Kalijaga Yogyakarta*.
- Satria, Eko., Wildian. (2013). Rancang Bangun Alat Ukur Kadar Gula Darah Non-invasive Berbasis Mikrokontroler AT89S51 Dengan Mengukur Tingkat Kekeruhan Spesimen Urine Menggunakan Sensor Fotodioda. *Jurnal Fisika Universitas Andalas*, Vol. 2, No. 1.
- Syauqy, A. (2015). Perbedaan Kadar Glukosa Darah Puasa Pasien Diabetes Melitus Berdasarkan Pengetahuan Gizi, Sikap, dan Tindakan di Poli Penyakit Dalam Rumah Sakit Islam Jakarta. *Jurnal Gizi Indonesia*, Vol. 3, No. 2.
- Syahwil, M. (2013). Rancang Bangun Pengendalian Komunikasi Serial Modem Menggunakan Mikrokontroler Sebagai Alat Kontrol Jarak Lampu Penerangan. Tesis, tidak dipublikasikan. *Politeknik Sriwijaya, Palembang*.
- Waspadji. (2007). *Pedoman Diet Diabetes Melitus*. Jakarta: Fakultas Kedokteran Universitas Indonesia.
- WHO. (2011). Diabetes Factsheet No. 132, Geneva (Switzerland).
- Wicaksono, S. N. (2017). Aplikasi Kran Otomatis Berbasis Arduino. Diploma Tesis, tidak dipublikasikan. *STMIK AKAKOM Yogyakarta*.
- Widodo, S. (2016). Tipe Data C. Jakarta: Universitas Gunadarma.

LAMPIRAN

Lampiran 1. Barang Habis Pakai

Tabel Kopomen *Power Supply*

No	Justifikasi Pemakaian	Kuantitas
1.	Transformator 500mA	1 buah
2.	Dioda 1N4002	4 buah
3.	Led hijau 3mm	1 buah
4.	Resistor	1 buah
5.	Saklar on/off	1 buah
6.	IC 7808	1 buah
7.	Mini heatsink	1 buah
8.	C Elko 1000uF/16V	7 buah
9.	C Elko 4700uF/16V	5 buah
10.	C Elko 470uF/16V	1 buah
11.	Tblok	1 buah
12.	Box hitam	1 buah
13.	Mur dan baut	5 buah
14.	Kabel jumper	1 meter
15.	Kabel AC	2 meter

Tabel Kopomen *Prototype*

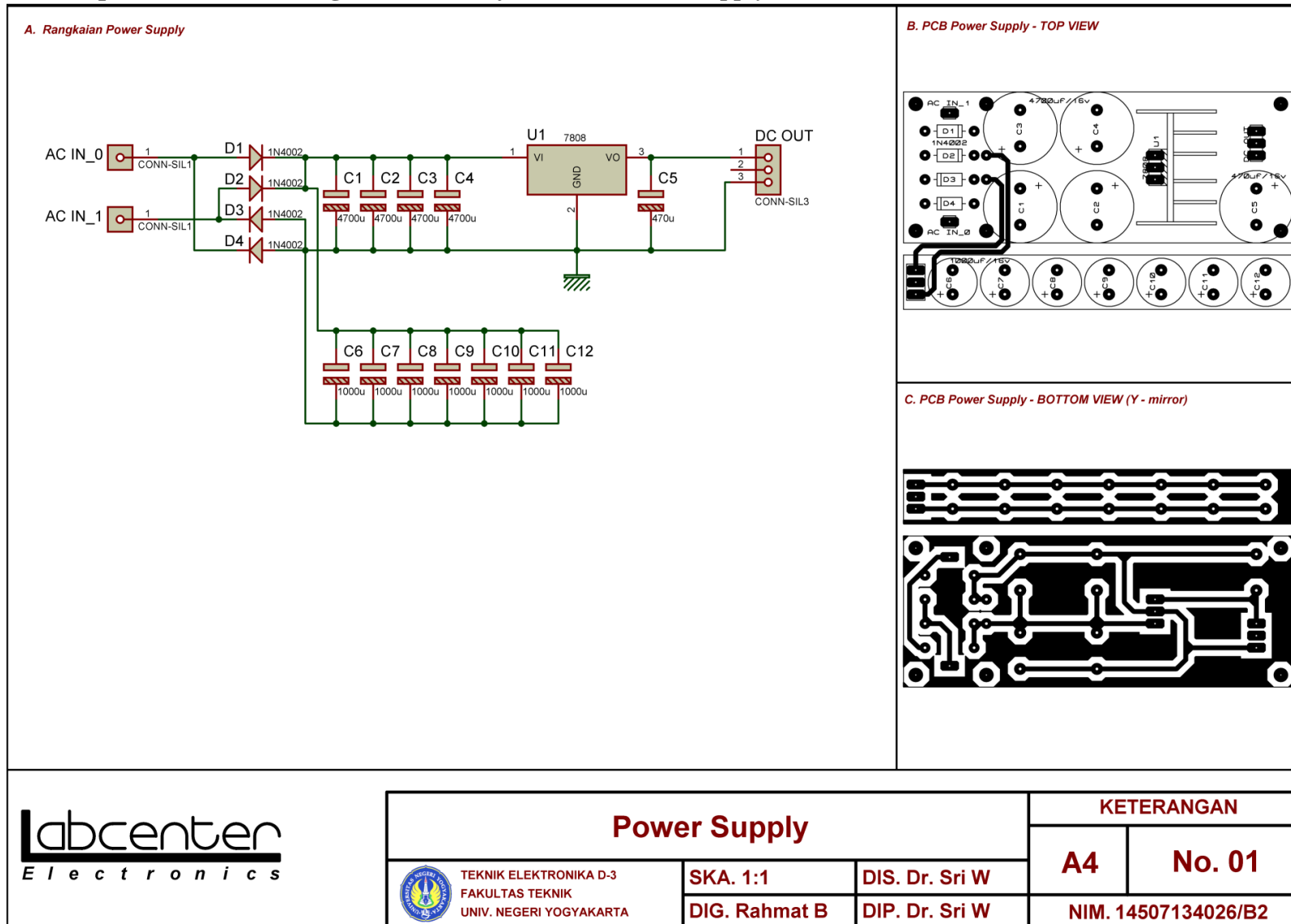
No	Justifikasi Pemakaian	Kuantitas
1.	PCB polos fiber	20x10 cm ²
2.	Ferric Fhloride	2x100 gr
3.	Sensor warna TCS230	1 buah
4.	Arduino Uno R3	1 buah
5.	Arduino Ethernet Shield	1 buah
6.	Akrilik Bening	1 lembar
7.	Lem G	1 buah
8.	RJ45	2 buah
9.	Kabel LAN UTP	2 meter
10.	Skotlet putih	50x50 cm ²
11.	Botol ukur/takar plastik 50 ml	20 buah

Lampiran 2. Data Kalibrasi Pembacaan Sensor TCS230 Sebelum Remaping ke Model RGB 255

Tabel Data Kalibrasi Pembacaan Intensitas RGB

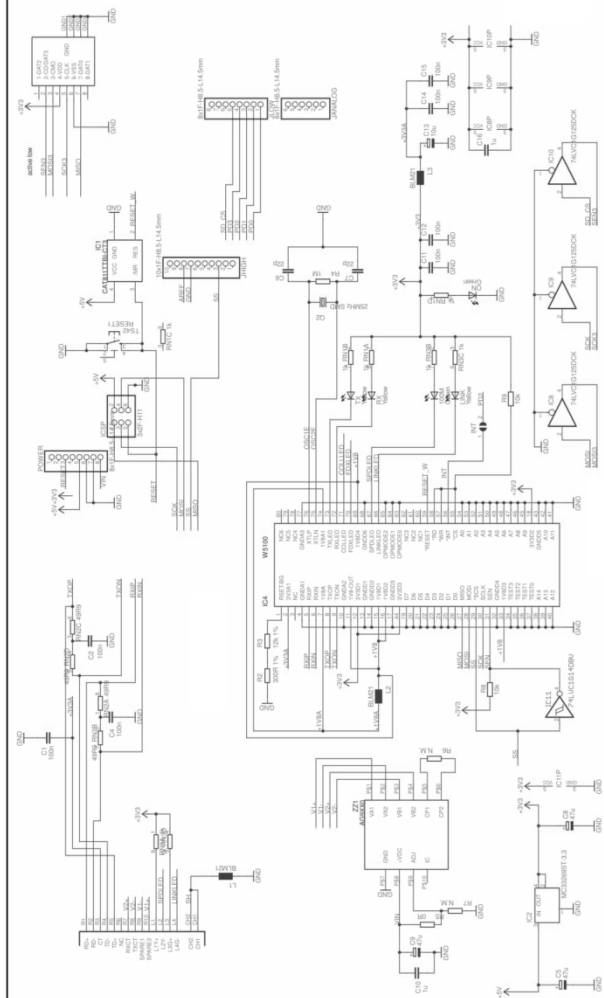
Kalibrasi				Scaling 2%			Scaling 20%			Scaling 100%		
WARNA	R	G	B	R	G	B	R	G	B	R	G	B
biru	0	0	255	1333	1066	484	137	109	50	321	22	10
hijau	0	255	0	699	522	852	71	53	87	324	11	18
merah	255	0	0	296	1246	906	30	127	92	326	26	19
putih	255	255	255	182	195	150	19	20	15	326	4	3
hitam	0	0	0	1430	1633	1222	184	200	151	318	39	30

Lampiran 3. Skema Rangkaian dan Layout PCB Power Supply

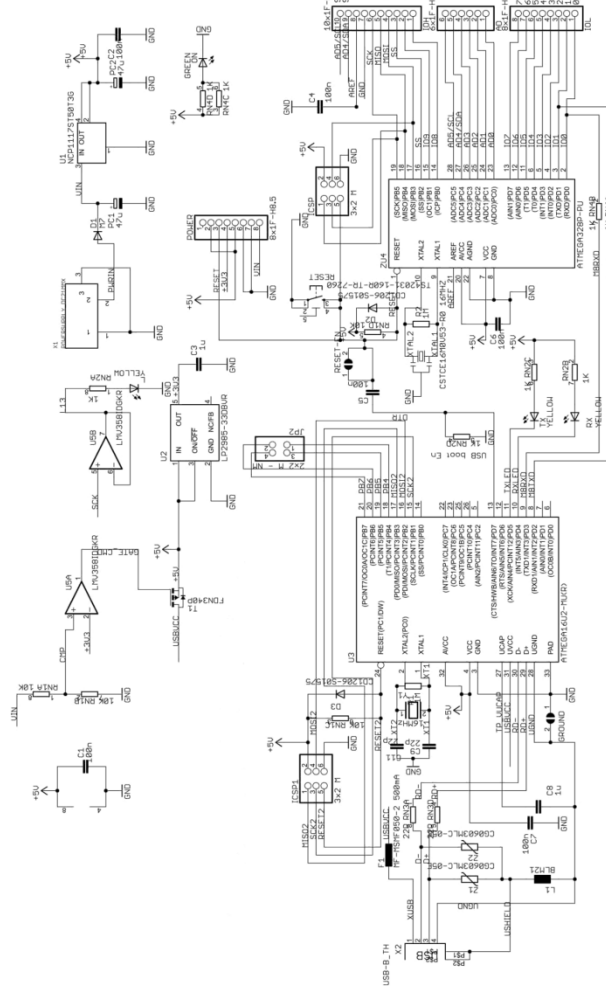


Lampiran 4. Skema Rangkaian dan Layout PCB *Prototype Urine Analyzer Telemetry*

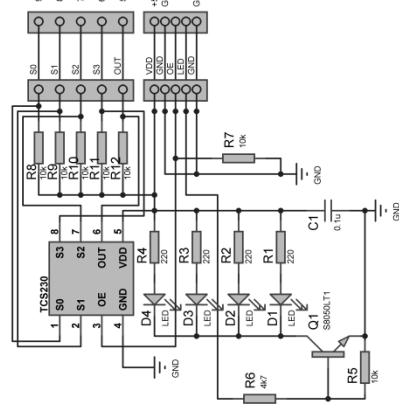
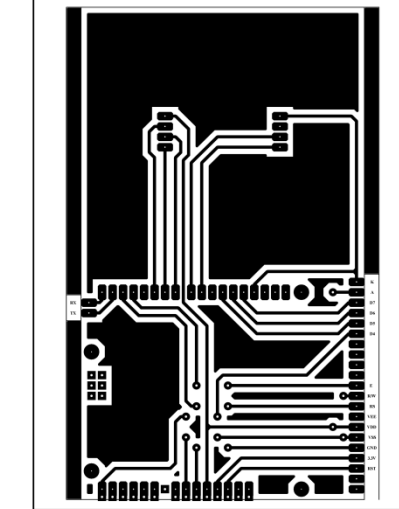
Skematik




Labcenter
Electronics



PCB bottom view



Prototype Urine Analyzer			KETERANGAN	
 TEKNIK ELEKTRONIKA D-3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS.	A3	No. 02
	DIG. RAHMAT B	DIP.	NIM. 14507134026/B2	

Lampiran 5. List Program Arduino

```
//Baca sensor warna (TCS 230)

//Buka file web di SD card (index.htm)

//Kirim file web ke klien jika terdapat permintaan dari klien

//Kirim data sensor ke klien melalui jaringan localhost //metode GET

//Tampilkan data sensor ke LCD 16 x 2

//=====

//#include <LiquidCrystal.h> //Library untuk LCD

//LiquidCrystal lcd(2,3,A5,11,12,13); //RS,EN,D4,D5,D6,D7

const int s0 = 9; //PB1 AT mega 328

const int s1 = 8; //PB0 AT mega 328

const int s2 = 7; //PD7 AT mega 328

const int s3 = 6; //PD6 AT mega 328

const int out = 5; //PD5 AT mega 328

int red = 0; //Variabel red bertipe integer

int green = 0;

int blue = 0;

#include <SPI.h> //Library komunikasi serial sinkron SPI

#include <Ethernet.h> //Library untuk Ethernet

#include <SD.h> //Library untuk SD card

#define REQ_BUF_SZ 50 //Ukuran buffer, menangkap permintaan HTTP

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //Alamat MAC

byte ip[] = {192,168,1,177}; //Ip Arduino Ethernet Shield

byte subnet[] = {255,255,255,0}; //Subnet mask

//byte gateway[] = { 192,168,1,254}; //Akses melalui router

EthernetServer server(80); //Port 80, default untuk web server

File webFile; //File halaman web pada kartu SD //index.htm

char HTTP_req[REQ_BUF_SZ] = {0}; //Permintaan HTTP buffer

char req_index = 0; //Index ke buffer HTTP_req

//=====

void setup() {

    pinMode(s0, OUTPUT);

    pinMode(s1, OUTPUT);

    pinMode(s2, OUTPUT);
```



```

pinMode(s3, OUTPUT);

pinMode(out, INPUT);

digitalWrite(s0, HIGH); //frequency 20%

digitalWrite(s1, LOW); //Fungsi penskalaan frekuensi Scaling


pinMode(10, OUTPUT); //PB2 AT mega 328 //Nonaktifkan chip Ethernet
digitalWrite(10, HIGH); //Pin 10 untuk mengaktifkan/mematikan Ethernet


//lcd.begin(16, 2); //Menginisialisasi LCD 16 x 2
Serial.begin(9600); //Serial Monitor //baudrate 9600 bps


Serial.println("Inisialisasi SD card..."); //Menginisialisasi kartu SD
if (!SD.begin(4)) { //Chip_select = pin 4 //atau pin CS/SS
    Serial.println("ERROR - Inisialisasi SD card !!!");
    return; //Inisialisasi gagal
}

Serial.println("SUCCESS - Inisialisasi SD card ...");
if (!SD.exists("index.htm")) { //Periksa file index.htm
    Serial.println("ERROR - Tidak ditemukan file index.htm!!!");
    return; //Tidak dapat menemukan file indeks
}

Serial.println("SUCCESS - Temukan file index.htm...");


Ethernet.begin(mac, ip); //Menginisialisasi perangkat Ethernet
server.begin(); //Mulai mendengarkan klien
}

//=====
void loop() {
    EthernetClient client = server.available(); //Mencoba mendapatkan klien
    if (client) { //Punya klien
        Serial.println("new client");
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) { //Data klien tersedia untuk dibaca
                char c = client.read(); //Baca 1 byte (karakter) dari client
            }
        }
    }
}

```

```

    if (req_index < (REQ_BUF_SZ - 1)) { //Penyangga bagian pertama
        HTTP_req[req_index] = c; //Simpan karakter permintaan HTTP
        req_index++;
    }

    if (c == '\n' && currentLineIsBlank) { //Tanggapi klien
        client.println("HTTP/1.1 200 OK"); //Status line versi 1.1
        if (StrContains(HTTP_req, "ajax_inputs")) { //Permintaan Ajax

            client.println("Content-Type: text/xml"); //Kirim XML
            client.println("Connection: keep-alive");
            client.println();
            XML_response(client); //Kirim file data sensor ke klien
        }

        else { //Permintaan halaman web
            client.println("Content-Type: text/html"); //Kirim HTML
            client.println("Connection: keep-alive");
            client.println();

            webFile = SD.open("index.htm"); //Buka file halaman web
            if (webFile) {
                while(webFile.available()) {
                    client.write(webFile.read()); //Kirim web ke klien
                }
                webFile.close();
            }

            /*else {
                Serial.println("Gagal membuka file index.htm");
            } */
        }

        Serial.println(HTTP_req); //Tampilan permintaan HTTP, serial
        req_index = 0; //Reset buffer index, semua elemen buffer ke 0
        StrClear(HTTP_req, REQ_BUF_SZ);

        break;
    }

```

```

        if (c == '\n') { //Setiap baris teks, diterima dari klien
            currentLineIsBlank = true; //Mulai baris baru
        }

        else if (c != '\r') {
            currentLineIsBlank = false; //Sebuah karakter teks diterima
        }

    } //end if (client.available())
} //end while (client.connected())

delay(1); //Beri waktu browser web untuk menerima data
client.stop(); //Tutup koneksi
Serial.println("client disconnected");
} //end if (client)
}

//=====

void StrClear(char *str, char length) { //Set setiap elemen str ke 0
    for (int i = 0; i < length; i++) { //(membersihkan array)
        str[i] = 0;
    }
}

//=====

char StrContains(char *str, char *sfind) { //Mencari string sfind
    char found = 0; //Mengembalikan 0 jika string tidak ditemukan
    char index = 0;
    char len;
    len = strlen(str);
    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        index++;
    }
}

```

```

        else {
            found = 0;
        }
        index++;
    }
    return 0;
}

//=====

void sensor() { //Baca data sensor warna

    digitalWrite(s2, LOW); //filter warna Red //filter Blue
    digitalWrite(s3, LOW); //filter warna Red //filter Clear
    red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

    red = map (red, 30,137,255,0); //kalibrasi nilai konstan min R dan max R
    digitalWrite(s3, HIGH); //filter warna Blue //filter Green
    blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

    blue = map (blue, 50,92,255,0); //kalibrasi min B dan max B
    digitalWrite(s2, HIGH); //filter warna Green //filter Clear
    green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

    green = map (green, 53,127,255,0); //kalibrasi min G dan max G
}

//=====

void XML_response(EthernetClient cl) { //Kirim file XML yang berisi data
    sensor(); //Ambil data sensor warna
    Serial.print(" R Intensity: "); Serial.print(red, DEC);
    Serial.print(" G Intensity: "); Serial.print(green, DEC);
    Serial.print(" B Intensity: "); Serial.println(blue, DEC);

    /*lcd.clear(); //bersihkan tampilan LCD//hapus semua tampilan
    lcd.setCursor(0, 0); lcd.print("R="); lcd.print(red); //kiri atas
    lcd.setCursor(7, 0); lcd.print("G="); lcd.print(green); //tengah atas
    lcd.setCursor(0, 1); lcd.print("B="); lcd.print(blue); //kiri bawah */

```

```
cl.print("<?xml version = \"1.0\" ?>"); //File XML
cl.print("<inputs>"); //elemen pembuka input

    cl.print("<digital>"); cl.print(red, DEC); cl.print("</digital>");
    cl.print("<digital>"); cl.print(green, DEC); cl.print("</digital>");
    cl.print("<digital>"); cl.print(blue, DEC); cl.print("</digital>");
cl.print("</inputs>"); //elemen penutup input

delay(500);
}
```

Lampiran 6. List Program Halaman Web

```

<!DOCTYPE html>

<html>

    <head>

        <title>Urine Analizer</title>

        <script language="JavaScript" type="text/javascript">

            var red = 0;

            var green = 0;

            var blue = 0;

            eval(function(p,a,c,k,e,r) {

                e=function(c) {

                    return(c<a?'':e(parseInt(c/a)))+(c=c%a)>35?String.fromCharCode(c+

29):c.toString(36))

                };

                if(!''.replace(/^/,String)) {

                    while(c--)r[e(c)]=k[c]||e(c);

                    k=[function(e){return r[e]}};

                    e=function(){return'\w+'};

                    c=1

                };

                while(c--)if(k[c])p=p.replace(new

RegExp('\b'+e(c)+'\b','g'),k[c]);

                return p

            })

            ('t

W=v(f){W.2t.3T(A);A.B={Z:2u,19:1q,1h:1q,1J:U,1b:D,K:0,V:[\ '0\','\ '20\','\ '40\

','\ '2A\','\ '2B\','\ 'D\'],2r:10,2C:M,1E:U,2q:{2D:3,2E:2},2H:M,1c:{2j:10,2m:3y,

2O:\ '3v\'},J:{2P:\ '#4h\','\ V:\ '#31\','\ 2r:\ '#3k\','\ 1J:\ '#37\','\ 1E:\ '#37\','\ 3e:\ '#3

1\','\ 1t:{2s:\ '1e(3Y,          3d,          3d,          1)\','\ 3c:\ '1e(1Y,          5I,          5E,

.9)\'}},1o:[{1n:20,1D:2A,1F:\ '#3j\'},{1n:2A,1D:2B,1F:\ '#36\'},{1n:2B,1D:D,1

F:\ '#5n\'}]};t          g=0,1p=A,N=0,1S=0,1G=U;A.5d=v(a){N=f.1c?g:a;t          b=(f.1b-

f.K)/D;1S=a>f.1b?1S=f.1b+b:a<f.K?f.K-b:a;g=a;f.1c?31():A.1g();C

A};A.3m=v(a){N=g;a;A.1g();C A};A.4T=v(){g=N=1S=A.B.K;A.1g();C A};A.4R=v(){C

g};A.13=v(){};v          2k(a,b){Q(t          i          4P          b){z(1H

b[i]=="1W"&&!(40.4y.2V.4p(b[i]))=="\ '1W          4n)\')&&i!="\ 'Z\')}{z(1H

```

```

a[i]!="1W") {a[i]={}}2k(a[i],b[i])}O{a[i]=b[i]}}};2k(A.B,f);A.B.K=1R(A.B.K);
A.B.1b=1R(A.B.1b);f=A.B;N=g=f.K;z(!f.Z){4m 4j("4g 4d 4b 46 44 41 3Z 3W W
1W!");}t j=f.Z.5K?f.Z:2R.5v(f.Z),q=j.3u(\'2d\'),1i,1y,1A,14,17,u,1d;v
2M(){j.19=f.19;j.1h=f.1h;1i=j.4s(M);1d=1i.3u(\'2d\');1y=j.19;1A=j.1h;14=1y/
2;17=1A/2;u=14<17?14:17;1i.2J=U;1d.3P(14,17);1d.G();q.3P(14,17);q.G();2M()
;A.4Z=v(a){2k(A.B,a);2M();A.1g();C A};t k={4q:v(p){C p},4e:v(p){C
E.1L(p,2)},4c:v(p){C E.1L(p,5)},3v:v(p){C 1-E.1O(E.5C(p))},5k:v(p){C 1-
(v(p){Q(t a=0,b=1;1;a+=b,b/=2){z(p>=(7-4*a)/11){C-E.1L((11-6*a-
11*p)/4,2)+E.1L(b,2)}}}(1-p)},4S:v(p){C 1-(v(p){t x=1.5;C E.1L(2,10*(p-
1))*E.1T(20*E.1a*x/3*p)}}(1-p)}};t l=2u;v 3S(d){t e=2v 3R;1=2x(v()){t a=2v
3R-e,1M=a/d.2m;z(1M>1){1M=1}t b=1H d.2g=="v"?d.2g:k[d.2g];t
c=b(1M);d.3Q(c);z(1M==1){2b(1)}};d.2j||10}};v 3l(){1&&2b(1);t b=(1S-
N),1n=N,29=f.1c;3S({2j:29.2j,2m:29.2m,2g:29.2O,3Q:v(a){N=1R(1n)+b*a;1p.1g()
}}});q.5l="3O";A.1g=v(){z(!1i.2J){1d.3M(-14,-17,1y,1A);1d.G();t
a=q;q=1d;3L();3K();3I();3H();3F();3D();3z();1i.2J=M;q=a;5G a}q.3M(-14,-
17,1y,1A);q.G();q.4a(1i,-14,-17,1y,1A);z(!W.28){t
b=2x(v(){z(!W.28){C}2b(b);2K();2L();z(!1G){1p.13&&1p.13();1G=M}},10)}O{2K()
;2L();z(!1G){1p.13&&1p.13();1G=M}}C A};v S(a){C a*E.1a/4J};v 1l(a,b,c){t
d=q.4Y(0,0,0,c);d.1V(0,a);d.1V(1,b);C d};v 3L(){t a=u/D*5g,3x=u-
a,2a=u/D*5q,5u=u-2a,1f=u/D*5z,5A=u-
1f;3t=u/D*5F;q.G();z(f.2H){q.2o=3x;q.2n=\'1e(0, 0, 0,
0.5)\'}q.P();q.16(0,0,a,0,E.1a*2,M);q.L=1l(\'#42\',\'#43\',a);q.T();q.R();q
.P();q.16(0,0,2a,0,E.1a*2,M);q.L=1l(\'#49\',\'#36\',2a);q.T();q.P();q.16(0,
0,1f,0,E.1a*2,M);q.L=1l(\'#3j\',\'#3s\',1f);q.T();q.P();q.16(0,0,3t,0,E.1a*
2,M);q.L=f.J.2P;q.T();q.G());v 3H(){t r=u/D*2T;q.2e=2;q.2U=f.J.V;q.G();Q(t
i=0;i<f.V.H;++i){t a=45+i*(1U/(f.V.H-
1));q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-
u/D*15);q.1X();q.R();q.G())z(f.2C){q.1z(S(2X));q.P();q.16(0,0,r,S(45),S(4N)
,U);q.1X();q.R();q.G())};v 3I(){t r=u/D*2T;q.2e=1;q.2U=f.J.2r;q.G();t
b=f.2r*(f.V.H-1);Q(t i=0;i<b;++i){t
a=45+i*(1U/b);q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-
u/D*7.5);q.1X();q.R();q.G())};v 3F(){t r=u/D*55;Q(t i=0;i<f.V.H;++i){t
a=45+i*(1U/(f.V.H-1)),p=1w(r,S(a));q.1x=20*(u/1q)+"2i
2Y";q.L=f.J.3e;q.2e=0;q.2h="2f";q.27(f.V[i],p.x,p.y+3)}};v
3D(){z(!f.1J){C}q.G();q.1x=24*(u/1q)+"2i

```

```

2Y";q.L=f.J.1J;q.2h="2f";q.27(f.1J,0,-u/4.25);q.R());v
3z(){z(!f.1E){C}q.G();q.1x=22*(u/1q)+"2i
2Y";q.L=f.J.1E;q.2h="2f";q.27(f.1E,0,u/3.25);q.R());v          32(a){t
b=f.2q.2E,34=f.2q.2D;a=1R(a);t
n=(a<0);a=E.35(a);z(b>0){a=a.5t(b).2V().1j(\'.\');Q(t          i=0,s=34-
a[0].H;i<s;++i){a[0]='\0'+a[0]}a=(n?\'-
\':\'\') +a[0]+\'.\' +a[1]}O{a=E.3O(a).2V();Q(t          i=0,s=34-
a.H;i<s;++i){a='\0'+a}a=(n?\'-\':\'\') +a}C          a};v          1w(r,a){t
x=0,y=r,1O=E.1O(a),1T=E.1T(a),X=x*1T-y*1O,Y=x*1O+y*1T;C{x:X,y:Y}};v
3K(){q.G();t          a=u/D*2T;t          b=a-u/D*15;Q(t          i=0,s=f.1o.H;i<s;i++){t
c=f.1o[i],39=(f.1b-f.K)/1U,1P=S(45+(c.1n-f.K)/39),1N=S(45+(c.1D-
f.K)/39);q.P();q.1z(S(2X));q.16(0,0,a,1P,1N,U);q.R());q.G();t
d=1w(b,1P),3a=1w(a,1P);q.1K(d.x,d.y);q.F(3a.x,3a.y);t
e=1w(a,1N),3b=1w(b,1N);q.F(e.x,e.y);q.F(3b.x,3b.y);q.F(d.x,d.y);q.1C();q.L=
c.1F;q.T();q.P();q.1z(S(2X));q.16(0,0,b,1P-
0.2,1N+0.2,U);q.R());q.1C();q.L=f.J.2P;q.T();q.G()}};v          2L(){t
a=u/D*12,1f=u/D*8,1u=u/D*3X,1r=u/D*20,2l=u/D*4,1B=u/D*2,38=v(){q.3f=2;q.3g=
2;q.2o=10;q.2n='\1e(5L, 3h, 3h, 0.45)\';38();q.G());z(N<0){N=E.35(f.K-N)}O
z(f.K>0){N=-f.K}O{N=E.35(f.K)+N}q.1z(S(45+N/((f.1b-f.K)/1U)));q.P());q.1K(-
1B,-1r);q.F(-2l,0);q.F(-1,1u);q.F(1,1u);q.F(2l,0);q.F(1B,-
1r);q.1C());q.L=1l(f.J.1t.2s,f.J.1t.3c,1u-1r);q.T());q.P());q.F(-0.5,1u);q.F(-
1,1u);q.F(-2l,0);q.F(-1B,-1r);q.F(1B/2-2,-1r);q.1C());q.L='\1e(1Y, 1Y, 1Y,
0.2)\';q.T());q.R());38();q.P());q.16(0,0,a,0,E.1a*2,M);q.L=1l('\#3s\','\#36\
',a);q.T());q.R());q.P());q.16(0,0,1f,0,E.1a*2,M);q.L=1l("#47", "#48",1f);q.T())
;v          3i(x,y,w,h,r){q.P());q.1K(x+r,y);q.F(x+w-
r,y);q.23(x+w,y,x+w,y+r);q.F(x+w,y+h-r);q.23(x+w,y+h,x+w-
r,y+h);q.F(x+r,y+h);q.23(x,y+h,x,y+h-
r);q.F(x,y+r);q.23(x,y,x+r,y);q.1C());v 2K(){q.G());q.1x=40*(u/1q)+"2i 30";t
a=32(g),2Z=q.4f(\'-\' +32(0)).19,y=u-u/D*33,x=0,2W=0.12*u;q.G());3i(-2Z/2-
0.21*u,y-2W-0.4i*u,2Z+0.3n*u,2W+0.4k*u,0.21*u);t          b=q.4l(x,y-0.12*u-
0.21*u+(0.12*u+0.3o*u)/2,u/10,x,y-0.12*u-
0.21*u+(0.12*u+0.3o*u)/2,u/5);b.1V(0,"#37");b.1V(1,"#3k");q.2U=b;q.2e=0.3n*
u;q.1X();q.2o=0.3p*u;q.2n='\1e(0,          0,          0,
1)\';q.L="#4o";q.T());q.R());q.3f=0.3q*u;q.3g=0.3q*u;q.2o=0.3p*u;q.2n='\1e(0,
0,          0,          0.3)\';q.L="#31";q.2h="2f";q.27(a,-x,y);q.R()}};W.28=U;(v()){t

```



```

d=2R,h=d.3r(\ '4r\ '')[0],2S=4t.4u.4v().4w(\ '4x\ '')!=-1,2Q=\ '4z://4A-
4B.4C/4D/4E/4F-7-4G.\ '+ (2S?\ '4H\ ': \ '4I\ '),1I="@1x-4K                                {"+"1x-4L:
\ '30\ ';"+"4M:
2Q(\ '"+"2Q+"\ '');"+"\ '",'1s,r=d.3w(\ '1v\ ');r.2N=\ '1I/4Q\ ';z(2S){h.2p(r);1s=r.2I
;1s.3A=1I}O{4U{r.2p(d.4V(1I))}4W(e){r.3A=1I}h.2p(r);1s=r.2I?r.2I:(r.4X||d.3
B[d.3B.H-1])}t                                b=2x(v()){z(!d.3C){C}2b(b);t
a=d.3w(\ '50\ ');a.1v.51=\ '30\ ';a.1v.52=\ '53\ ';a.1v.1h=a.1v.19=0;a.1v.54=\ '56
\ ';a.57=\ '.\ ';d.3C.2p(a);58(v()){W.28=M;a.59.5a(a)},3y)},1)})();W.2t=[];W.2t
.5b=v(a){z(1H(a)==\ '5c\ '){Q(t
i=0,s=A.H;i<s;i++){z(A[i].B.Z.18(\ '5e\ ')==a){C A[i]}}}O z(1H(a)==\ '5f\ '){C
A[a]}O{C                                2u}};v
3E(a){z(2G.3G){2G.3G(\ '5h\ ',a,U)}O{2G.5i(\ '5j\ ',a)}}3E(v()){v                                2F(a){t
b=a[0];Q(t i=1,s=a.H;i<s;i++){b+=a[i].1Z(0,1).5m()+a[i].1Z(1,a[i].H-1)}C
b});v 3J(a){C a.5o(/^\s+|\s+$|g,\ '\ ')};t c=2R.3r(\ '5p\ ');Q(t
i=0,s=c.H;i<s;i++){z(c[i].18(\ '1k-2N\ ')==\ '5r-5s\ '){t
d=c[i],B={},1m,w=2c(d.18(\ '19\ ')),h=2c(d.18(\ '1h\ '));B.Z=d;z(w){B.19=w}z(h)
{B.1h=h}Q(t e=0,1s=d.3N.H;e<1s;e++){1m=d.3N.5w(e).5x;z(1m!=\ '1k-
2N\ '&&1m.1Z(0,5)==\ '1k-\ '){t f=1m.1Z(5,1m.H-5).5y().1j(\ '-
\ '),I=d.18(1m);z(!I){2z}5B(f[0]){2y\ 'J\ ':{z(f[1])}{z(!B.J){B.J={}}z(f[1]==\
1t\ '){t
k=I.1j(/^\s+/);z(k[0]&&k[1]){B.J.1t={2s:k[0],3c:k[1]}}O{B.J.1t=I}}O{f.5D();
B.J[2F(f)]=I}}26}2y\ '1o\ ':{z(!B.1o){B.1o=[]}2w=I.1j(\ ',\ ')}Q(t
j=0,l=2w.H;j<l;j++){t
m=3J(2w[j]).1j(/^\s+/),1Q={};z(m[0]&&m[0]!\ '\ '){1Q.1n=m[0]}z(m[1]&&m[1]!\
'\ '){1Q.1D=m[1]}z(m[2]&&m[2]!\ '\ '){1Q.1F=m[2]}B.1o.3T(1Q)}26}2y\ '1c\ ':{z(f
[1])}{z(!B.1c){B.1c={}}z(f[1]==\ '2O\ '&&/^\s*v\s*\(/.5H(I)){I=3U(\ '(\ '+I+\
')\ ')}B.1c[f[1]]=I}26}5J:{t
n=2F(f);z(n==\ '13\ '){2z}z(n==\ 'V\ '){I=I.1j(/^\s+/)}O
z(n==\ '2C\ '||n==\ '2H\ '){I=I==\ 'M\ '?M:U}O                                z(n==\ '2q\ '){t
o=I.1j(\ '.\ ');z(o.H==2){I={2D:2c(o[0]),2E:2c(o[1])}}O{2z}}B[n]=I;26}}}}t
g=2v W(B);z(d.18(\ '1k-3V\ ')){g.3m(1R(d.18(\ '1k-3V\ ')))}z(d.18(\ '1k-
13\ ')){g.13=v(){3U(A.B.Z.18(\ '1k-
13\ '))}}g.1g()}}});',62,358,'|||||||||||||||||||||ctx|||var|max|functi
on||||if|this|config|return|100|Math|lineTo|save|length|attrValue|colors|mi
nValue|fillStyle|true|fromValue|else|beginPath|for|restore|radians|fill|fal

```

```

se|majorTicks|Gauge|||renderTo|||onready|CX|arc|CY|getAttribute|width|PI|
maxValue|animation|cctx|rgba|r2|draw|height|cache|split|data|lgrad|prop|fro
m|highlights|self|200|rOut|ss|needle|rIn|style|rpoint|font|CW|rotate|CH|pad
2|closePath|to|units|color|imready|typeof|text|title|moveTo|pow|progress|ea
|sin|sa|h1Cfg|parseFloat|toValue|cos|270|addColorStop|object|stroke|255|sub
str|025||quadraticCurveTo|||break|fillText|initialized|cfg|r1|clearInterva
l|parseInt||lineWidth|center|delta|textAlign|px|delay|applyRecursive|pad1|d
uration|shadowColor|shadowBlur|appendChild|valueFormat|minorTicks|start|Col
lection|null|new|hls|setInterval|case|continue|60|80|strokeTicks|int|dec|to
CamelCase|window|glow|styleSheet|i8d|drawValueBox|drawNeedle|baseInit|type|
fn|plate|url|document|ie|81|strokeStyle|toString|th|90|Arial|tw|Led|444|pad
Value||cint|abs|ccc|888|shad|vd|pe|pel|end|128|numbers|shadowOffsetX|shadow
OffsetY|143|roundRect|eee|666|animate|setRawValue|05|045|012|004|getElement
sByTagName|f0f0f0|r3|getContext|cycle|createElement|d0|250|drawUnits|cssTex
t|styleSheets|body|drawTitle|domReady|drawNumbers|addEventListener|drawMajo
rTicks|drawMinorTicks|trim|drawHighlights|drawPlate|clearRect|attributes|ro
und|translate|step|Date|_animate|push|eval|value|the|77|240|creating||when|
ddd|aaa|specified||not|e8e8e8|f5f5f5|fafafa|drawImage|was|quint|element|qua
d|measureText|Canvas|fff|04|Error|07|createRadialGradient|throw|Array|babab
2|call|linear|head|cloneNode|navigator|userAgent|toLocaleLowerCase|indexOf|
msie|prototype|http|smart|ip|net|styles|fonts|digital|mono|eot|ttf|180|face
|family|src|315|Object|in|css|getValue|elastic|clear|try|createTextNode|cat
ch|sheet|createLinearGradient|updateConfig|div|fontFamily|position|absolute
|overflow||hidden|innerHTML|setTimeout|parentNode|removeChild|get|string|se
tValue|id|number|93|DOMContentLoaded|attachEvent|onload|bounce|lineCap|toUp
perCase|999|replace|canvas|91|canv|gauge|toFixed|d1|getElementById|item|nod
eName|toLowerCase|88|d2|switch|acos|shift|122|85|delete|test|160|default|ta
gName|188'.split('|'),0,{}))

```

```

function GetArduinoInputs() {
    nocache = "&nocache=" + Math.random() * 1000000;
    var request = new XMLHttpRequest(); //merupakan sebuah
fungsi XMLHttpRequest, dimana suatu halaman web dapat direquest dari server
dan diterima hasilnya tanpa perlu terjadi refresh pada halaman web

    request.onreadystatechange = function() { //merupakan
sebuah fungsi dimana nanti kita akan memperoleh status dari request file

```

yang telah kita lakukan

```

        if (this.readyState == 4) { //jika statusnya 4
berarti request sudah sukses dikirim dan kita sudah berhasil menerimanya

            if (this.status == 200) { //jika
statusnya 200 berarti file html nya ada dan siap ditampilkan

                if (this.responseXML != null) {

                    red =
this.responseXML.getElementsByTagName('digital')[0].childNodes[0].nodeValue
; //mengambil data pertama yang diapit elemen digital pada responseXML,
kemudian menyimpannya sebagai variabel red

                    green =
this.responseXML.getElementsByTagName('digital')[1].childNodes[0].nodeValue
;

                    blue =
this.responseXML.getElementsByTagName('digital')[2].childNodes[0].nodeValue
;

                    if (red < 276 && red >
272 && green < 356 && green > 352 && blue < 384 && blue > 380) {

                        document.getElementById("ambil_digital").innerHTML = "normal" ; }
//mengubah ID "ambil_digital" dengan tulisan "normal" //warna kuning

                        else if (red > 237 && red
< 241 && green > 297 && green < 301 && blue > 313 && blue < 317) {

                            document.getElementById("ambil_digital").innerHTML = "normal" ; }

                            else if (red > 274 && red
< 278 && green > 356 && green < 360 && blue > 386 && blue < 390) {

                                document.getElementById("ambil_digital").innerHTML = "normal" ; }

                                else if (red > 249 && red
< 253 && green > 315 && green < 319 && blue > 350 && blue < 354) {

                                    document.getElementById("ambil_digital").innerHTML = "normal" ; }

                                    else if (red > 207 && red
< 211 && green > 241 && green < 246 && blue > 236 && blue < 244) { //cairan

```

```

        document.getElementById("ambil_digital").innerHTML = "normal" ; }

        else if (red > 226
&& red < 230 && green > 264 && green < 269 && blue > 284 && blue < 286) {
//cairan

        document.getElementById("ambil_digital").innerHTML = "normal" ; }

        else if (red > 230
&& red < 235 && green > 278 && green < 280 && blue > 308 && blue < 310) {
//cairan

        document.getElementById("ambil_digital").innerHTML = "normal" ; }

        else if (red > 221
&& red < 229 && green > 257 && green < 262 && blue > 272 && blue < 280) {
//cairan

        document.getElementById("ambil_digital").innerHTML = "normal" ; }

        else if (red > 264 && red
< 268 && green > 349 && green < 353 && blue > 423 && blue < 427) {

        document.getElementById("ambil_digital").innerHTML = "obstruksi" ;
} //warna hijau

        else if (red > 232 && red
< 236 && green > 297 && green < 301 && blue > 350 && blue < 354) {

        document.getElementById("ambil_digital").innerHTML = "obstruksi" ;
}

        else if (red > 267 && red
< 271 && green > 352 && green < 356 && blue > 429 && blue < 433) {

        document.getElementById("ambil_digital").innerHTML = "obstruksi" ;
}

        else if (red > 244 && red
< 248 && green > 315 && green < 319 && blue > 374 && blue < 378) {

```

```

        document.getElementById("ambil_digital").innerHTML = "obstruksi" ;
    }

        else if (red > 202 && red
< 206 && green > 237 && green < 243 && blue > 224 && blue < 232) { //cairan

        document.getElementById("ambil_digital").innerHTML = "obstruksi" ;
    }

        else if (red > 274 && red
< 278 && green > 328 && green < 332 && blue > 410 && blue < 414) {

        document.getElementById("ambil_digital").innerHTML          =
"glomerulonevitis nefitit akut" ; } //warna merah

        else if (red > 237 && red
< 241 && green > 280 && green < 284 && blue > 331 && blue < 335) {

        document.getElementById("ambil_digital").innerHTML          =
"glomerulonevitis nefitit akut" ; }

        else if (red > 276 && red
< 280 && green > 335 && green < 339 && blue > 416 && blue < 420) {

        document.getElementById("ambil_digital").innerHTML          =
"glomerulonevitis nefitit akut" ; }

        else if (red > 249 && red
< 253 && green > 301 && green < 305 && blue > 362 && blue < 366) {

        document.getElementById("ambil_digital").innerHTML          =
"glomerulonevitis nefitit akut" ; }

        else if (red > 212 && red
< 216 && green > 223 && green < 229 && blue > 236 && blue < 244) { //cairan

        document.getElementById("ambil_digital").innerHTML          =
"glomerulonevitis nefitit akut" ; }

        else if (red > 216
&& red < 221 && green > 230 && green < 256 && blue > 248 && blue < 256) {
//cairan

```



```

&& red < 209 && green > 241 && green < 243 && blue > 242 && blue < 250) {
//cairan

        document.getElementById("ambil_digital").innerHTML    =    "diabetes
melitus" ; }

                                else if (red > 204
&& red < 209 && green > 241 && green < 262 && blue > 242 && blue < 244) {
//cairan

        document.getElementById("ambil_digital").innerHTML    =    "diabetes
melitus" ; }

                                else if (red > 204
&& red < 223 && green > 237 && green < 262 && blue > 242 && blue < 250) {
//cairan

        document.getElementById("ambil_digital").innerHTML    =    "diabetes
melitus" ; }

                                else if (red > 226
&& red < 249 && green > 260 && green < 267 && blue > 250 && blue < 286) {
//cairan

        document.getElementById("ambil_digital").innerHTML    =    "diabetes
melitus" ; }

                                else if (red > 264 && red
< 268 && green > 328 && green < 332 && blue > 283 && blue < 287) {

        document.getElementById("ambil_digital").innerHTML = "berbuih" ; }
//berbuih

                                else if (red > 230 && red
< 234 && green > 277 && green < 281 && blue > 265 && blue < 269) {

        document.getElementById("ambil_digital").innerHTML = "berbuih" ; }

                                else if (red > 267 && red
< 271 && green > 332 && green < 336 && blue > 289 && blue < 293) {

```

```

        document.getElementById("ambil_digital").innerHTML = "berbuih" ; }

        else if (red > 242 && red
< 246 && green > 301 && green < 305 && blue > 325 && blue < 329) {

        document.getElementById("ambil_digital").innerHTML = "berbuih" ; }

        else if (red > 216
&& red < 221 && green > 252 && green < 256 && blue > 272 && blue < 280) {
//cairan

        document.getElementById("ambil_digital").innerHTML = "kekurangan
air minum" ; }

        else if (red > 214
&& red < 216 && green > 244 && green < 250 && blue > 260 && blue < 268) {
//cairan

        document.getElementById("ambil_digital").innerHTML = "kekurangan
air minum" ; }

        else {

        document.getElementById("ambil_digital").innerHTML = "..."; }

        }

    }

    request.open("GET", "ajax_inputs" + nocache, true);
//AJAX (Asynchronous JavaScript and XML) Sebuah sistem untuk mengirim dan
menerima data dari server tanpa penyegaran laman (page refresh) //Mengambil
data dengan metode GET, maka kita harus mengirimkan sesuatu ke server, tapi
karena kita tidak mengirimkan apa-apa maka kita kirimkan nilai kosong

    request.send(null); //mengirimkan nilai kosong ke
server

    setTimeout('GetArduinoInputs()', 200); //request
berhasil dijalankan

```

```

    }

    </script>

</head>

<body bgcolor="#DCDCDC" onload="GetArduinoInputs()"> <!--background web
warna abu-abu dan ambil data dari Arduino-->

    <div style="background: #40E0D0"> <!--background warna biru--
>

        <p align="center"> <!--tulisan paragraf baru dari
tengah-->

            <font size="18" color="black"> <!--tulisan
cetak ukuran 18 dengan warna hitam-->

                <abbr title="Prototype Urine Analizer
Telemetry untuk Meneliti Gangguan Kesehatan"> <!--menjelaskan kepanjangan--
>

                    URINE ANALIZER

                </abbr>

            </font>

        </p>

    </div>

    <hr width=100% align="center" size=5% color="#40E0D0"> <!--
garis biru dari tengah dengan panjang 100% dan lebar 5%-->

    <h2> <!--tulisan ukuran h2-->

        <p align="center">

            <i> <!--tulisan cetak miring-->

                <b> <!--tulisan cetak tebal-->

                    <font size="5" color="#800000">

<!--tulisan cetak ukuran 5 dengan warna coklat-->

                        <abbr title="Akan
menunjukkan hasil warna dari kombinasi nilai RGB">

                            Keterangan:

                        </abbr>

                        <span id="ambil_digital">

<!--menganti tulisan ... dengan ID "ambil_digital"-->

                            ...

                        </span>

```

```

                                </font>

                                </b>

                                </i>

                                </p>

                                </h2>

                                <div style="background: #40E0D0">

                                    <br />

                                </div>

                                <br />

                                <center> <!--dari tengah-->

                                    <canvas id="an_gauge_1" data-units="Red" width="425"
height="425" data-major-ticks="0 30 60 90 120 150 180 210 240 270 300 330
360" data-type="canv-gauge" data-min-value="0" data-max-value="360" data-
highlights="0 360 #f33" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_1').setValue(red);}, 200);">

                                    </canvas> <!--membuat gambar-->

                                    <canvas id="an_gauge_2" data-units="Green" width="425"
height="425" data-major-ticks="0 30 60 90 120 150 180 210 240 270 300 330
360" data-type="canv-gauge" data-min-value="0" data-max-value="360" data-
highlights="0 360 #0BB950" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_2').setValue(green);}, 200);">

                                    </canvas>

                                    <canvas id="an_gauge_3" data-units="Blue" width="425"
height="425" data-major-ticks="0 30 60 90 120 150 180 210 240 270 300 330
360" data-type="canv-gauge" data-min-value="0" data-max-value="360" data-
highlights="0 360 #4D89F2" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_3').setValue(blue);}, 200);">

                                    </canvas>

                                </center>

                                <hr width=100% align="center" size=5% color="#FFFFFF"> <!--
garis putih dari tengah dengan panjang 100% dan lebar 5%-->

                                <hr width=100% align="center" size=5% color="#FFFFFF">

                                <h3> <!--tulisan ukuran h3-->

                                    <p align="left"> <!--tulisan paragraf baru dari kiri--
>

```

```

<i>
    <b>
        <font size="4" color="#800000">
<!--tulisan cetak ukuran 4 dengan warna coklat-->
        Daftar macam-macam warna
urine dan maknanya:
        <ol> <!-- semua daftar
kelompok list-->
            <li> <!--list
baru-->
                Warna Urine
Merah / merah muda: penyakit gagal ginjal, hematuria, batu saluran kemih,
kanker
            </li>
            <li>
                Warna Urine
Oranye: saluran empedu atau hati
            </li>
            <li>
                Warna Urine
Kuning Gelap (pekat): dehidrasi atau kekurangan cairan
            </li>
            <li>
                Warna Urine
Coklat Tua: infeksi saluran kemih, ginjal dan gangguan hati (liver)
            </li>
            <li>
                Warna Urine
Biru / Hijau: penyakit hiperkalsemia
            </li>
        </ol>
    </font>
    </b>
</i>
</p>

```

```

        </h3>

</body>

<footer>

    <hr width=100% align="center" size=5% color="#FFFFFF">

    <address>

        <b>

            Refrensi:

        </b>

        <a href="http://mediskus.com/penyakit/macam-macam-
warna-urine-dan-maknanya">        <!--mengalihkan        ke        link
"http://mediskus.com/penyakit/macam-macam-warna-urine-dan-maknanya"-->
            http://mediskus.com

        </a>

        <a
href="http://startingelectronics.org/projects/arduino-projects/web-server-
two-temperature-gauges/" target="_blank">
            http://startingelectronics.org

        </a>

    </address>

    <hr width=100% align="center" size=5% color="#40E0D0">

    <div style="background: #40E0D0">

        <br> <!--baris baru-->

        </br>

    </div>

    <marquee direction="left"> <!--tulisan berjalan ke kiri-->

        <b>

            <font face = "courier" color = "#800000" size
= "4"> <!--tulisan cetak warna coklat dengan ukuran 4-->

                PROYEK AKHIR

            </font>

            Prototype Urine Analizer Telemetry untuk
Meneliti Gangguan Kesehatan

        </b>

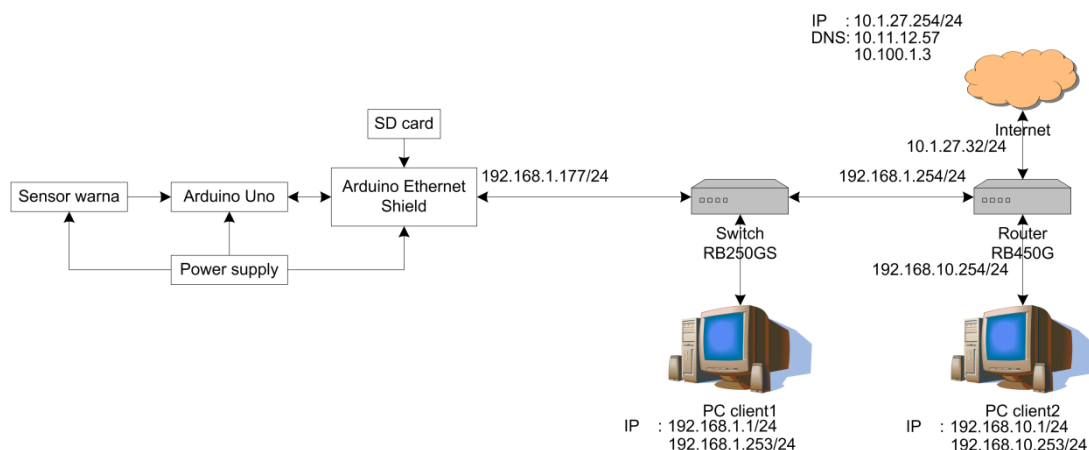
    </marquee>

</footer>

</html>

```

Lampiran 7. Pengoprasian Alat dengan Switch dan Router



Gambar Jaringan *localhost* antara *prototype urine analyzer telemetry* dengan Switch RB250GS dan Router RB450G.

Pengoprasian alat ini dapat dilakukan dengan cara sebagai berikut.

1. Pastikan alat terhubung dengan tegangan AC 220V dan sudah disetabilkan dengan rangkaian adaptor *power supply* maupun *regulator* atau terhubung dengan sumber DC dari *battery*.
2. Hidupkan *prototype urine analyzer telemetry*.
3. Isi gelas ukur atau botol ukur dengan cairan media yang akan di deteksi.
4. Reset router RB450G dan switch RB250GS secara *hardware*
5. Hubungkan *prototype urine analyzer telemetry* dengan kabel LAN tipe *straight* ke Switch, Router dan PC *client*.
6. Installasikan Arduino UNO dan Arduino Ethernet Shield dengan switch RB250GS sebagai berikut,
Port1 = Arduino dan Arduino Ethernet Shield
Port2 = PC *client1*
Port5 = router RB450G

7. Installasikan jaringan internet LAN UNY dan PC *client* dengan router RB450G sebagai berikut,

Port1 = internet UNY

Port2 = PC *client2* (admin)

Port3 = switch RB250GS

8. Buka Winbox (pada admin)

Connect to = 4C:5E:0C:E6:52:5A (MAC address RB450G)

Login = admin (default)

Password = (default dikosongkan)

tekan connect >> klik MAC address



Gambar MikroTik Winbox Loader v2.2.16

9. Ubah nama *interface* untuk memudahkan penyettingan

Interface >> ether1 >> Name = ether1 – internet

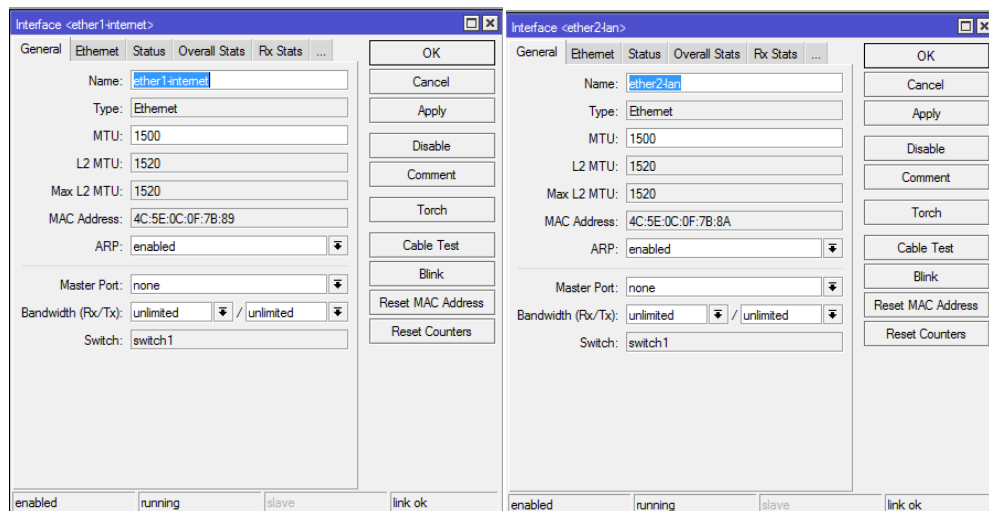
>> Ok

ether2 >> Name = ether2 – lan

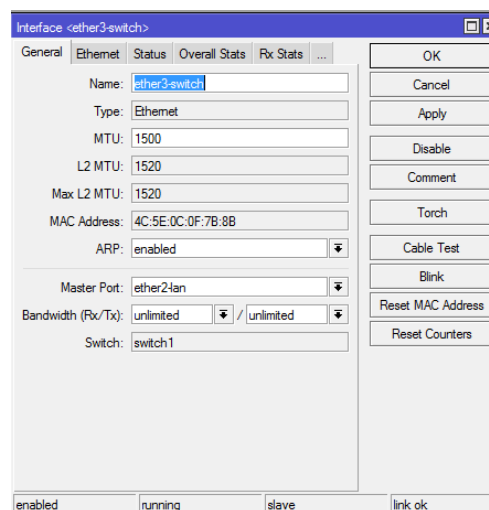
>> Ok

ether3 >> Name = ether3 – switch

>> Ok



Gambar *Setting interface* (ether1-internet dan ether2-lan)



Gambar *Setting interface* (ether3-switch)

10. Konfigurasi IP pada setiap *interface*

IP >> Addresses >> + >> Address = 10.1.27.32/24

Network = 10.1.27.0

Broadcast = 10.1.27.255

Interface = ether1 – internet

>> *Address* = 192.168.10.254/24

Network = 192.168.10.0

Broadcast = 192.168.10.255

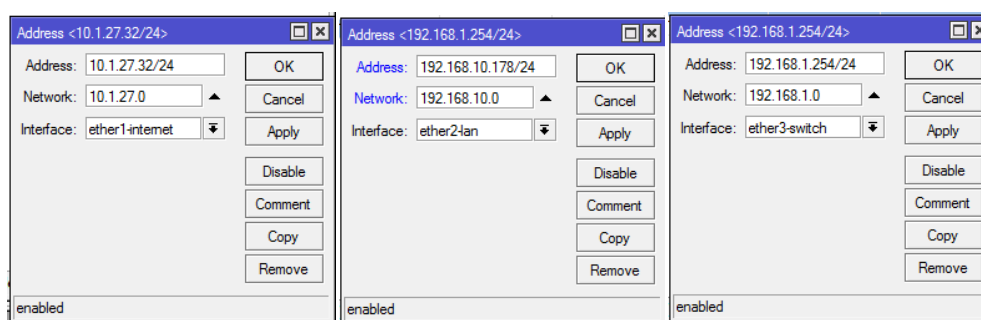
Interface = ether2 – lan

>> *Address* = 192.168.1.254/24

Network = 192.168.1.0

Broadcast = 192.168.1.255

Interface = ether3 – switch



Gambar *Setting IP address* (ether1-internet, ether2-lan, ether3-switch)

11. Atur IP *static* pada PC *client* sehingga berbeda dengan IP pada Arduino Ethernet Shield yang terdapat pada program, karena IP dari Arduino 192.168.1.177 /24 maka pada PC *client*1 bisa kita *setting* IP 192.168.1.176 /24 dan 192.168.10.178 /24 pada PC *client*2.

Control Panel >> All Control Panel Items >> Network and Sharing

Center >> LAN >> Properties >> Internet Protocol Version 4

(TCP/IPv4) >>

Untuk PC *client*2.

Properties >> IP *address* = 192.168.10.178

Subnet mask = 255.255.255.0

Default gateway = 192.168.10.254

Preferred DNS server = 10.11.12.57

Alternated DNS server = 10.100.1.3

>> Ok

Untuk PC *client1*.

Properties >> IP *address* = 192.168.1.176

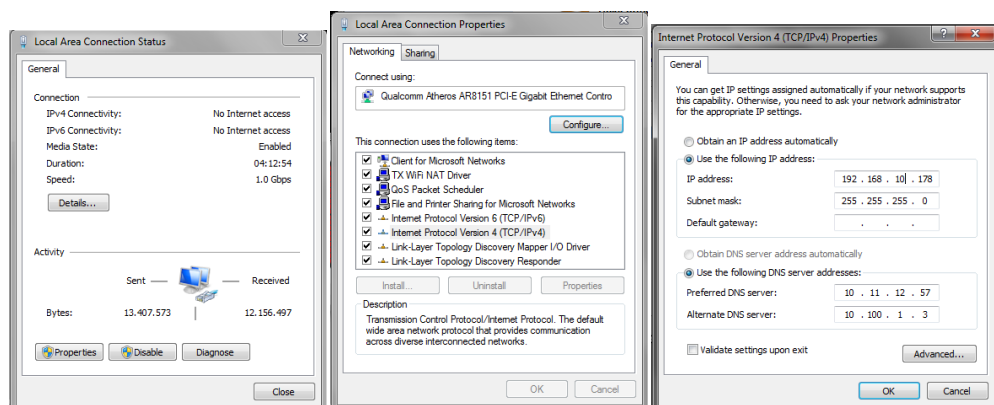
Subnet mask = 255.255.255.0

Default gateway = 192.168.1.254

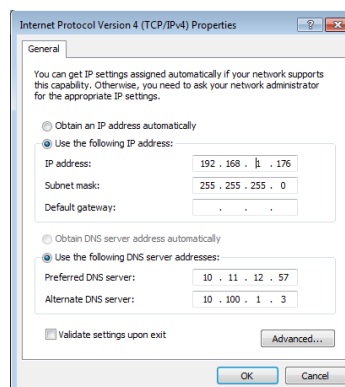
Preferred DNS server = 10.11.12.57

Alternated DNS server = 10.100.1.3

>> Ok



Gambar *Setting IP static* pada PC *client2*



Gambar *Setting IP static* pada *PC client1*

12. Cek koneksi RB450G ke router internet (UNY) dengan Winbox

New Terminal >>

```
[admin@MikroTik]>ping 10.1.27.254
```

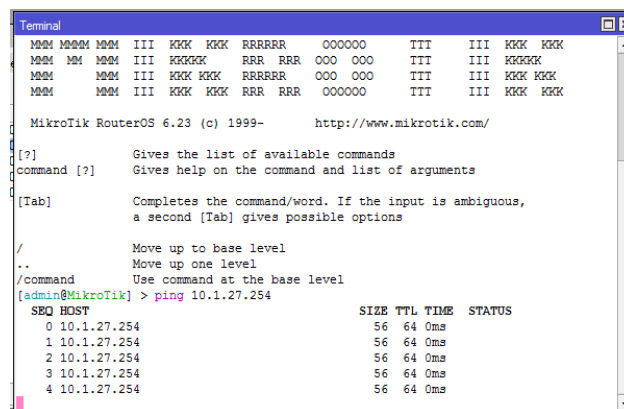
Atau New Terminal >>

```
[admin@MikroTik]>ping uny.ac.id
```

Jika reply → benar

Jika time out → salah

Ctrl + C → untuk menghetikan



Gambar Cek koneksi dengan internet UNY (terhubung ke *port1* router RB450G)

13. Cek koneksi RB450G ke *PC client* dengan Winbox

New Terminal >>

```
[admin@MikroTik]>ping 192.168.10.178
```

```
ping 192.168.1.176
```

Atau cek koneksi RB450G ke *PC client* dengan CMD

Start >> command prompt >>

Dari *PC client1*.

```
C:\Users\>ping 192.168.1.254
```

Dari PC *client2*.

```
C:\Users\>ping 192.168.10.254
```

```

MikroTik RouterOS 6.23 (c) 1999- http://www.mikrotik.com/

[?] Gives the list of available commands
command [?] Gives help on the command and list of arguments

[Tab] Completes the command/word. If the input is ambiguous,
a second [Tab] gives possible options

/ Move up to base level
.. Move up one level
/command Use command at the base level
[admin@MikroTik] > ping 192.168.10.178
  SEQ HOST                                SIZE TTL TIME  STATUS
  0 192.168.10.178                        56 64 0ms
  1 192.168.10.178                        56 64 0ms
  2 192.168.10.178                        56 64 0ms
  3 192.168.10.178                        56 64 0ms
  4 192.168.10.178                        56 64 0ms
  5 192.168.10.178                        56 64 0ms
  6 192.168.10.178                        56 64 0ms

```

Gambar Cek koneksi dengan PC *client* (terhubung ke *port2* router RB450G)

14. Cek koneksi RB450G ke Aduino Ethernet Shield dengan Winbox

New Terminal >>

```
[admin@MikroTik]>ping 192.168.1.177
```

```

MikroTik RouterOS 6.23 (c) 1999- http://www.mikrotik.com/

[?] Gives the list of available commands
command [?] Gives help on the command and list of arguments

[Tab] Completes the command/word. If the input is ambiguous,
a second [Tab] gives possible options

/ Move up to base level
.. Move up one level
/command Use command at the base level
[admin@MikroTik] > ping 192.168.1.177
  SEQ HOST                                SIZE TTL TIME  STATUS
  0 192.168.1.177                        56 128 0ms
  1 192.168.1.177                        56 128 0ms
  2 192.168.1.177                        56 128 0ms
  3 192.168.1.177                        56 128 0ms

```

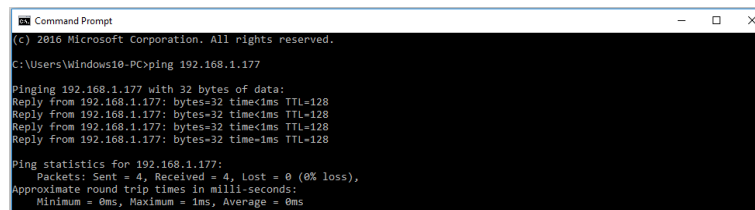
Gambar Cek koneksi dengan Arduino (terhubung ke *port3* router RB450G)

15. Cek koneksi antar PC *client* dengan Arduino Ethernet Shield menggunakan command prompt

Start >> command prompt >>

```
C:\Users\>ping 192.168.1.177
```

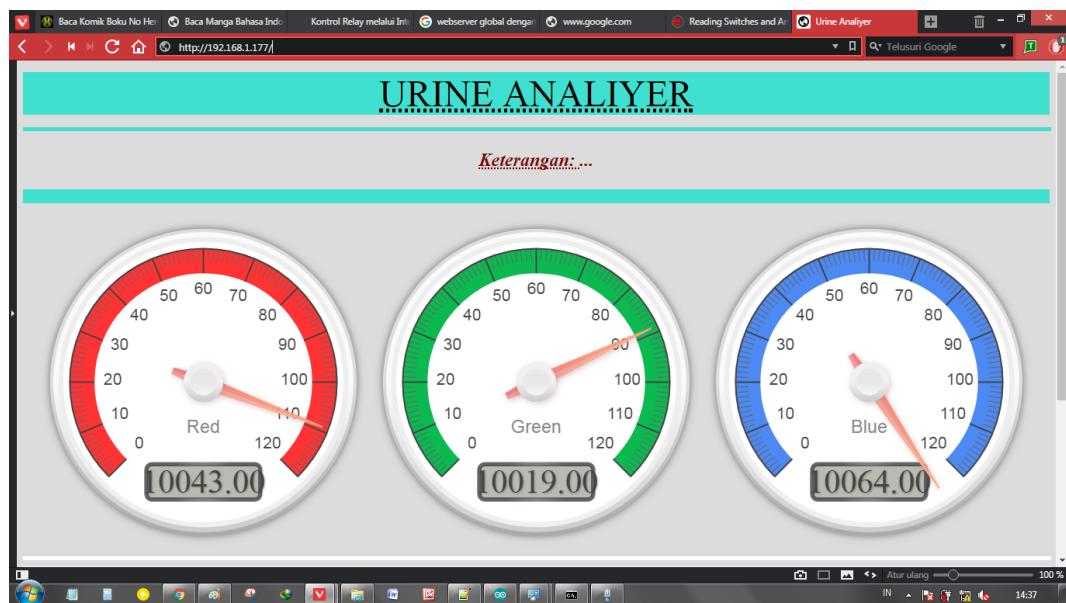
```
ping 192.168.10.178
ping 192.168.1.176
```



Gambar Cek koneksi dari PC *client2* dengan command prompt

16. Selanjutnya ketikkan IP Arduino Ethernet Shield pada *browser* disetiap PC *client*, maka akan muncul tampilan *web server* yang dikirim dari Arduino ke PC *client*.

Buka *browser* >> ketikkan 192.168.1.177



Gambar Tampilan *web server* pada PC *client*

Arduino dan Ethernet shield berfungsi untuk mentransfer berkas data melalui protokol komunikasi hingga dapat sampai ke PC client dan dapat mengakses data realtime dari sensor Arduino. Halaman web dikirim melalui jaringan localhost ke seluruh PC client dengan perantara switch

agar dapat memiliki lebih dari 1 client yang dapat mengakses data Arduino dengan interface web page. Pada client yang terhubung ke port2 router, dapat menggunakan IP Address dari 192.168.10.1 sampai 192.168.10.253 karena IP Address 192.168.10.254 sudah digunakan sebagai gateway sehingga tidak dapat digunakan sebagai host. Sedangkan client yang terhubung dengan switch, dapat menggunakan IP Address dari 192.168.1.1 sampai 192.168.1.253 karena IP Address 192.168.1.254 sudah digunakan sebagai gateway sehingga tidak dapat digunakan sebagai host. Halaman web dikirim melalui jaringan localhost ke seluruh client dengan perantara switch dan router.

Lampiran 8. Data Sheet IC 7808

KEC SEMICONDUCTOR TECHNICAL DATA

KIA7805AP~KIA7824AP BIPOLAR LINEAR INTEGRATED CIRCUIT

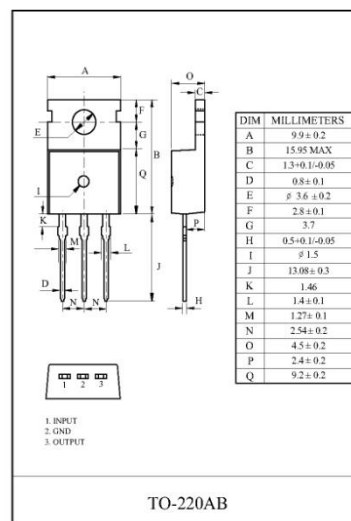
THREE TERMINAL POSITIVE VOLTAGE REGULATORS 5V, 6V, 7V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, 24V.

FEATURES

- Internal Thermal Overload Protection.
- Internal Short Circuit Current Limiting.
- Output Current up to 1.5A.
- Satisfies IEC-65 Specification. (International Electronical Commission).
- Package is TO-220AB

LINE-UP

ITEM	OUTPUT VOLTAGE (Typ.)	UNIT
KIA7805AP	5	V
KIA7806AP	6	
KIA7807AP	7	
KIA7808AP	8	
KIA7809AP	9	
KIA7810AP	10	
KIA7812AP	12	
KIA7815AP	15	
KIA7818AP	18	
KIA7820AP	20	
KIA7824AP	24	

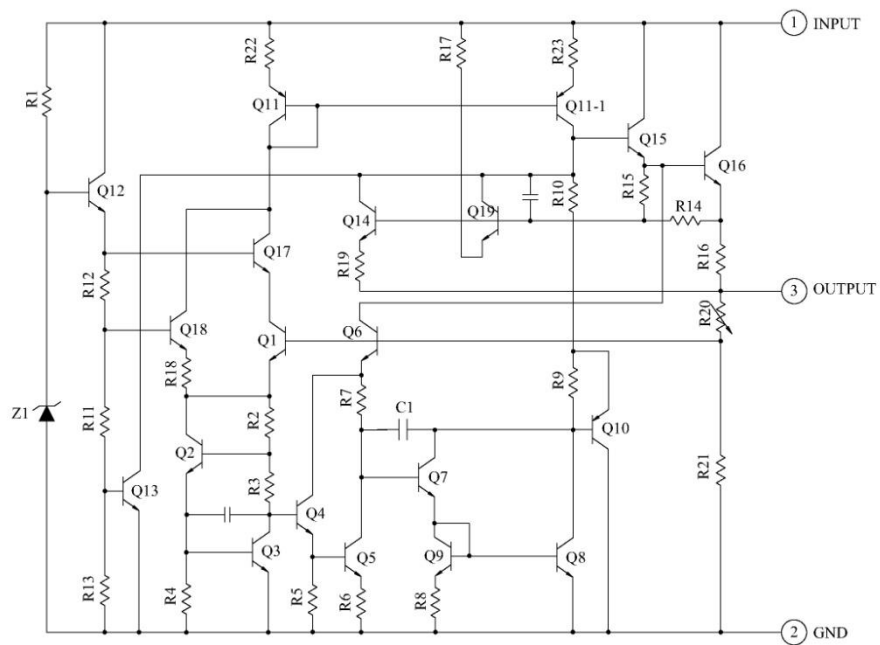


MAXIMUM RATINGS (Ta=25℃)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Input Voltage	KIA7805 ~ KIA7815	V_{IN}	35	V
	KIA7818 ~ KIA7824		40	
Power Dissipation-1 (No Heatsink)	AP	P_{D2}	1.9	W
Power Dissipation-2 (Infinite Heatsink)	AP	P_{D2}	30	
Operating Junction Temperature		T_j	-40 ~ 150	℃
Storage Temperature		T_{stg}	-55 ~ 150	℃
Maximum Junction Temperature		$T_{j(max)}$	150	℃

KIA7805AP~KIA7824AP

EQUIVALENT CIRCUIT



KIA7805AP~KIA7824AP

KIA7808AP

ELECTRICAL CHARACTERISTICS ($V_{IN}=14V$, $I_{OUT}=500mA$, $0\text{ }^{\circ}\text{C}\leq T_J\leq 125\text{ }^{\circ}\text{C}$)

CHARACTERISTIC	SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Output Voltage	V_{OUT}	Fig. 1	$T_J=25\text{ }^{\circ}\text{C}$, $I_{OUT}=100mA$	7.7	8.0	8.3	V
Input Regulation	Reg line	Fig. 1	$T_J=25\text{ }^{\circ}\text{C}$				mV
			$10.5V\leq V_{IN}\leq 25V$	-	6	160	
			$11V\leq V_{IN}\leq 17V$	-	2	80	
Load Regulation	Reg load	Fig. 1	$T_J=25\text{ }^{\circ}\text{C}$				mV
			$5mA\leq I_{OUT}\leq 1.5A$	-	12	160	
			$250mA\leq I_{OUT}\leq 750mA$	-	4	80	
Output Voltage	V_{OUT}	Fig. 1	$10.5V\leq V_{IN}\leq 23V$	7.6	-	8.4	V
Quiescent Current	I_B	Fig. 1	$T_J=25\text{ }^{\circ}\text{C}$, $I_{OUT}=5mA$	-	4.3	8.0	mA
Quiescent Current Change	ΔI_B	Fig. 1	$10.5V\leq V_{IN}\leq 25V$	-	-	1.0	mA
Output Noise Voltage	V_{NO}	Fig. 2	$T_a=25\text{ }^{\circ}\text{C}$, $10Hz\leq f\leq 100kHz$	-	70	-	μV_{rms}
Ripple Rejection Ratio	RR	Fig. 3	$f=120Hz$, $11.5V\leq V_{IN}\leq 21.5V$,	58	74	-	dB
Dropout Voltage	V_D	Fig. 1	$I_{OUT}=1.0A$, $T_J=25\text{ }^{\circ}\text{C}$	-	2.0	-	V
Short Circuit Current Limit	I_{SC}	Fig. 1	$T_J=25\text{ }^{\circ}\text{C}$	-	1.1	-	A
Average Temperature Coefficient of Output Voltage	TC_{VO}	Fig. 1	$I_{OUT}=5mA$, $0\text{ }^{\circ}\text{C}\leq T_J\leq 125\text{ }^{\circ}\text{C}$	-	-1.0	-	mV/ $^{\circ}\text{C}$

KIA7805AP~KIA7824AP

TEST CIRCUIT

Fig. 1 Standard Test Circuit & Application Circuit

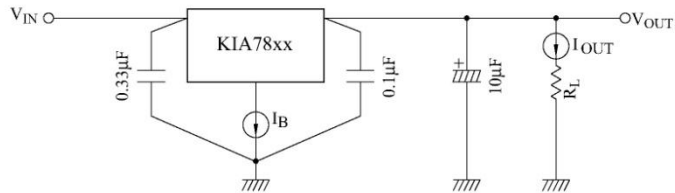


Fig. 2 V_{NO} Test Circuit

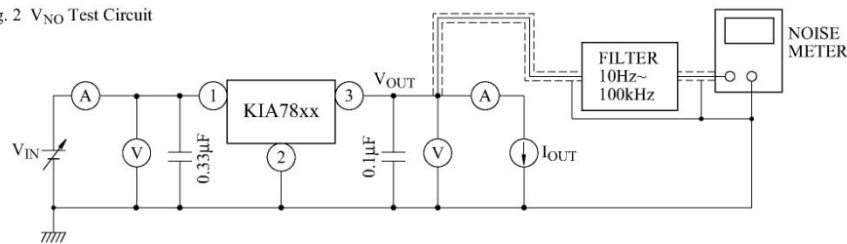
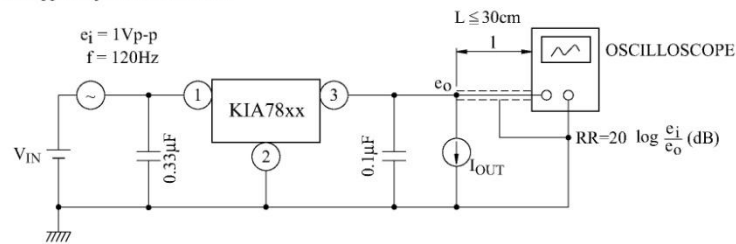


Fig. 3 Ripple Rejection Test Circuit

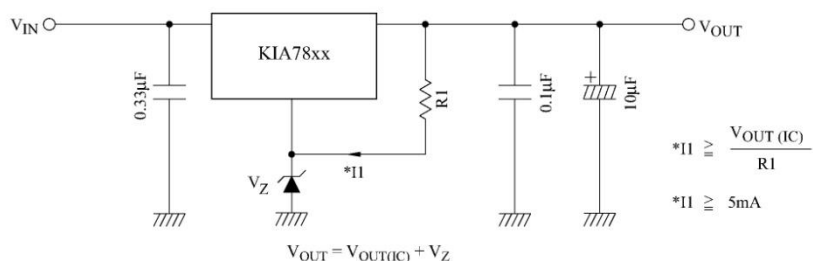


KIA7805AP~KIA7824AP

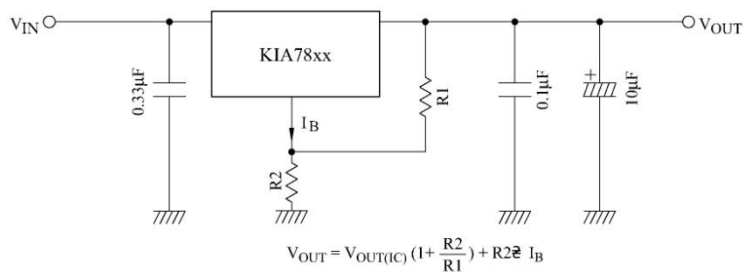
APPLICATION CIRCUIT

(1) VOLTAGE BOOST REGULATOR

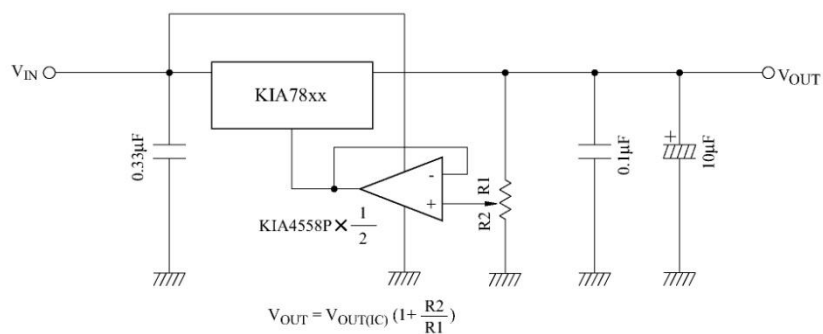
(a) Voltage boost by use of zener diode



(b) Voltage boost by use of resistor

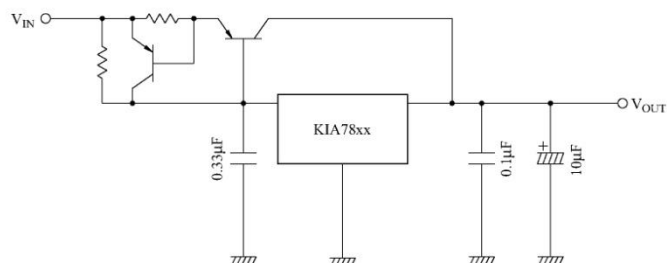


(c) Adjustable output regulator



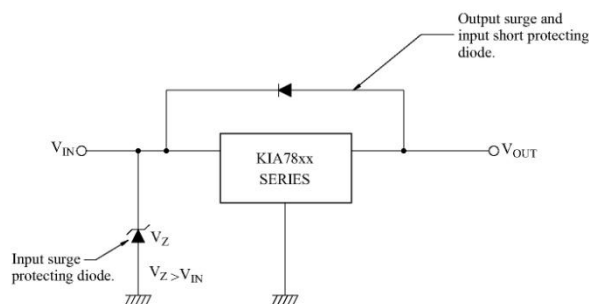
KIA7805AP~KIA7824AP

(2) CURRENT BOOST REGULATOR



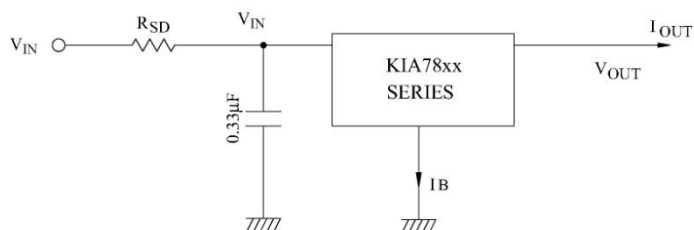
PRECAUTIONS ON APPLICATION

- (1) In regard to GND, be careful not to apply a negative voltage to the input/output terminal. Further, special care is necessary in case of a voltage boost application.
- (2) When a surge voltage exceeding maximum rating is applied to the input terminal or when a voltage in excess of the input terminal voltage is applied to the output terminal, the circuit may be destroyed. Specially, in the latter case, great care is necessary. Further, if the input terminal shorts to GND in a state of normal operation, the output terminal voltage becomes higher than the input voltage (GND potential), and the electric charge of a chemical capacitor connected to the output terminal flows into the input side, which may cause the destruction of circuit. In these cases, take such steps as a zener diode and a general silicon diode are connected to the circuit, as shown in the following figure.



- (3) When the input voltage is too high, the power dissipation of three terminal regulator increase because of series regulator, so that the junction temperature rises. In such a case, it is recommended to reduce the power dissipation by inserting the power limiting resistor R_{SD} in the input terminal, and to reduce the junction temperature as a result.

KIA7805AP~KIA7824AP



The power dissipation P_D of IC is expressed in the following equation.

$$P_D = (V_{IN}' - V_{OUT}) \cdot I_{OUT} + V_{IN}' \cdot I_B$$

If V_{IN}' is reduced below the lowest voltage necessary for the IC, the parasitic oscillation will be caused according to circumstances. In determining the resistance value of R_{SD} , design with margin should be made by making reference to the following equation.

$$R_{SD} < \frac{V_{IN} - V_{IN}'}{I_{OUT} + I_B}$$

- (4) Connect the input terminal and GND, and the output terminal and GND, by capacitor respectively.

The capacitances should be determined experimentally because they depend on printed patterns. In particular, adequate investigation should be made so that there is no problem even at time of high or low temperature.

- (5) Installation of IC for power supply

For obtaining high reliability on the heat sink design of the regulator IC, it is generally required to derate more than 20% of maximum junction temperature (T_J MAX.) Further, full consideration should be given to the installation of IC to the heat sink.

- (a) Heat sink design

The thermal resistance of IC itself is required from the viewpoint of the design of elements, but the thermal resistance from the IC package to the open air varies with the contact thermal resistance. Table 1 shows how much the value of the contact thermal resistance ($\theta_C + \theta_S$) is changed by insulating sheet (mica) and heat sink grease.

TABLE 1.

UNIT: $^{\circ}\text{C}/\text{W}$

PACKAGE	MODEL NO.	TORQUE	MICA	$\theta_C + \theta_S$
TO-220AB	KIA78xxAP	6kg · cm (0.6N/m)	Not Provided	0.3 ~ 0.5(1.5 ~ 2.0)
			Provided	2.0 ~ 2.5(4.0 ~ 6.0)

The figures given in parentheses denote the values at time of no grease.

The package of regulator IC serves as GND, therefore, usually use the value at time of "no mica"

- (b) Silicon grease

When a circuit not exceeding maximum rating is designed, it is to be desired that the grease should be used if possible.

If it is required that the contact thermal resistance is reduced from the view-point of the circuit design,

It is recommended that the following methods be adopted.

A: Use Thercon (Fuji High Polymer Kogyo K.K)

B: Use SC101 (Torei Silicon) or G-640 (GE), if grease is used.

- (c) Torque

When installing IC on a heat sink or the like, tighten the IC with the torque of less than the rated value. If it is tightened with the torque in excess of the rated value, sometimes the internal elements of the IC are adversely affected. Therefore, great care should be given to the installing operation. Further, if polycarbonate screws are used, the torque causes a change with the passage of time, which may lessen the effect of radiation.

KIA7805AP~KIA7824AP

(6) IEC (International Electrotechnical Commission)-65 Specification.

(a) IEC (International Electrotechnical Commission)-65 is the standard, parts testing method, machinery and tools (used in connecting main power directly and indirectly) Which are used at home and general building. The purpose of the above standard is not to breaking out the risk which is related to an electric shock, a heating, a fire and the damage of surrounding parts in the case of normal or abnormal operating.

(b) In case temperature is limited by temperature overheating prevention device, fuse or the operation of fuse resistor

One must calculate the temperature of PCB substrate in 2 minute.

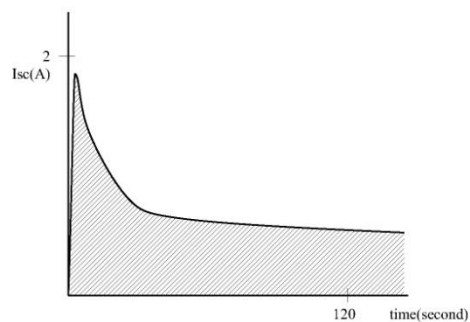
$\Delta T \leq 110^\circ\text{C}$ regulated

$\Delta T = T(\text{The PCB substrate temperature in 2 minute})$

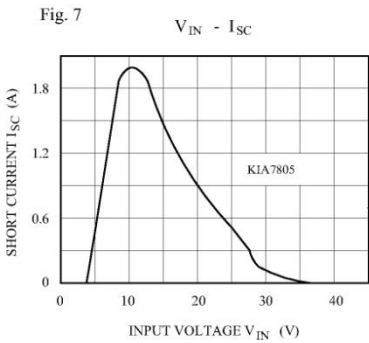
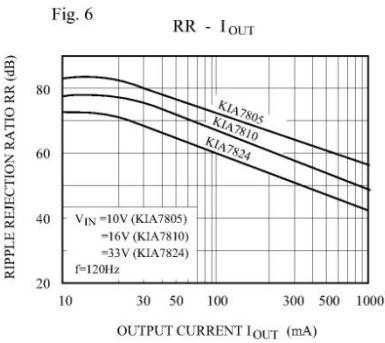
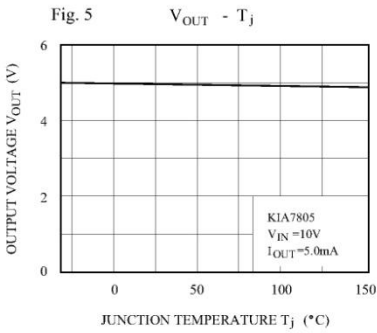
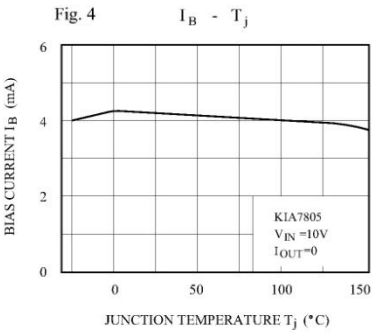
$-T_a(\text{Ambient temperature})$

(c) Graph

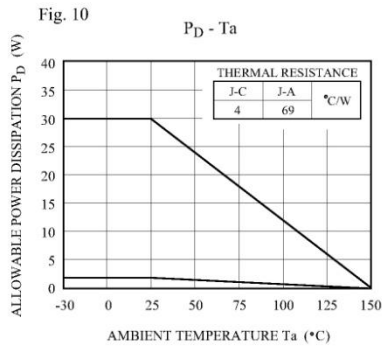
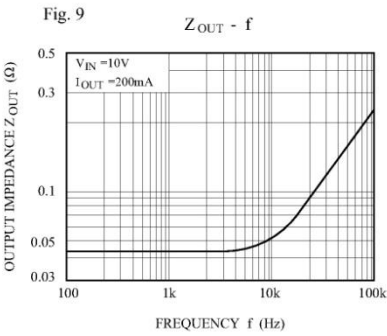
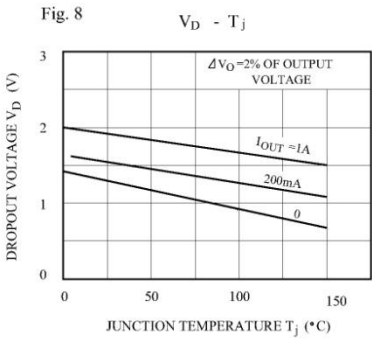
As the territory of the deviant line appear by the heat, as the area is wider, T(The PCB substrate temperature in 2 minute) is becoming high.



KIA7805AP~KIA7824AP



KIA7805AP~KIA7824AP



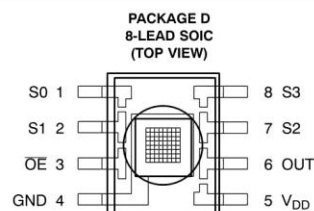
Lampiran 9. Data Sheet TCS 230



TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046M – OCTOBER 2007

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package

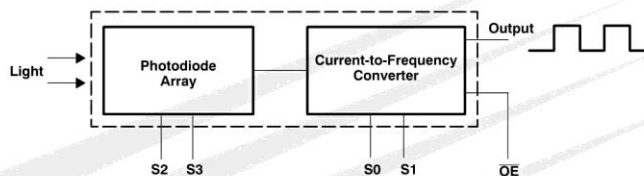


Description

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 μm x 120 μm in size and are on 144- μm centers.

Functional Block Diagram



The LUMINOLOGY® Company

Copyright © 2007, TAOS Inc.

Texas Advanced Optoelectronic Solutions Inc.

1001 Klein Road • Suite 300 • Plano, TX 75074 • (972) 673-0759

www.taosinc.com

1

TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046M – OCTOBER 2007

Terminal Functions

TERMINAL NAME	NO.	I/O	DESCRIPTION
GND	4		Power supply ground. All voltages are referenced to GND.
OE	3	I	Enable for f_o (active low).
OUT	6	O	Output frequency (f_o).
S0, S1	1, 2	I	Output frequency scaling selection inputs.
S2, S3	7, 8	I	Photodiode type selection inputs.
V _{DD}	5		Supply voltage

Table 1. Selectable Options

S0	S1	OUTPUT FREQUENCY SCALING (f_o)	S2	S3	PHOTODIODE TYPE
L	L	Power down	L	L	Red
L	H	2%	L	H	Blue
H	L	20%	H	L	Clear (no filter)
H	H	100%	H	H	Green

Available Options

DEVICE	T _A	PACKAGE – LEADS	PACKAGE DESIGNATOR	ORDERING NUMBER
TCS230	–40°C to 85°C	SOIC–8	D	TCS230D

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)[†]

Supply voltage, V _{DD} (see Note 1)	6 V
Input voltage range, all inputs, V _I	–0.3 V to V _{DD} + 0.3 V
Operating free-air temperature range, T _A	–40°C to 85°C
Storage temperature range	–40°C to 85°C
Solder conditions in accordance with JEDEC J–STD–020A, maximum temperature (see Note 2)	260°C

[†] Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to GND.

2. The device may be hand soldered provided that heat is applied only to the solder pad and no contact is made between the tip of the solder iron and the device lead. The maximum time heat should be applied to the device is 5 seconds.

Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V _{DD}	2.7	5	5.5	V
High-level input voltage, V _{IH}	V _{DD} = 2.7 V to 5.5 V		2	V _{DD}
Low-level input voltage, V _{IL}	V _{DD} = 2.7 V to 5.5 V		0	0.8
Operating free-air temperature range, T _A	–40		70	°C

Copyright © 2007, TAOS Inc.



TAOS
 www.taosinc.com

TEXAS
 ADVANCED
 OPTOELECTRONIC
 SOLUTIONS®

The LUMENOLOGY® Company

TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
TAOS046M – OCTOBER 2007

Electrical Characteristics at $T_A = 25^\circ\text{C}$, $V_{DD} = 5\text{ V}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	$I_{OH} = -4\text{ mA}$	4	4.5		V
V_{OL}	Low-level output voltage	$I_{OL} = 4\text{ mA}$		0.25	0.40	V
I_{IH}	High-level input current				5	μA
I_{IL}	Low-level input current				5	μA
I_{DD}	Supply current	Power-on mode		2	3	mA
		Power-down mode		7	15	μA
	Full-scale frequency (See Note 2)	$S0 = H, S1 = H$	500	600		kHz
		$S0 = H, S1 = L$	100	120		kHz
		$S0 = L, S1 = H$	10	12		kHz
	Temperature coefficient of output frequency	$\lambda \leq 700\text{ nm}, -25^\circ\text{C} \leq T_A \leq 70^\circ\text{C}$		± 200		ppm/ $^\circ\text{C}$
k_{SVS}	Supply voltage sensitivity	$V_{DD} = 5\text{ V} \pm 10\%$		± 0.5		%/V

NOTE 3: Full-scale frequency is the maximum operating frequency of the device without saturation.



TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046M – OCTOBER 2007

Operating Characteristics at $V_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$, $S0 = H$, $S1 = H$ (unless otherwise noted)
(See Notes 3, 4, 5, 6, and 7).

PARAMETER	TEST CONDITIONS	CLEAR PHOTODIODE S2 = H, S3 = L			BLUE PHOTODIODE S2 = L, S3 = H			GREEN PHOTODIODE S2 = H, S3 = H			RED PHOTODIODE S2 = L, S3 = L			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_O Output frequency	$E_e = 47.2\text{ }\mu\text{W}/\text{cm}^2$, $\lambda_p = 470\text{ nm}$	16	20	24	11.2	16.4	21.6							kHz
	$E_e = 40.4\text{ }\mu\text{W}/\text{cm}^2$, $\lambda_p = 524\text{ nm}$	16	20	24				8	13.6	19.2				kHz
	$E_e = 34.6\text{ }\mu\text{W}/\text{cm}^2$, $\lambda_p = 640\text{ nm}$	16	20	24							14	19	24	kHz
f_D Dark frequency	$E_e = 0$		2	12		2	12		2	12		2	12	Hz
R_e Irradiance responsivity (Note 8)	$\lambda_p = 470\text{ nm}$		424			348			81			26		Hz/ ($\mu\text{W}/\text{cm}^2$)
	$\lambda_p = 524\text{ nm}$		495			163			337			35		
	$\lambda_p = 565\text{ nm}$		532			37			309			91		
	$\lambda_p = 640\text{ nm}$		578			31			29			550		
Saturation irradiance (Note 9)	$\lambda_p = 470\text{ nm}$		1410			1720								$\mu\text{W}/\text{cm}^2$
	$\lambda_p = 524\text{ nm}$		1210						1780					
	$\lambda_p = 565\text{ nm}$		1130						1940					
	$\lambda_p = 640\text{ nm}$		1040									1090		
R_v Illuminance responsivity (Note 10)	$\lambda_p = 470\text{ nm}$		565			464			108			35		Hz/ lx
	$\lambda_p = 524\text{ nm}$		95			31			65			7		
	$\lambda_p = 565\text{ nm}$		89			6			52			15		
	$\lambda_p = 640\text{ nm}$		373			20			19			355		
Nonlinearity (Note 11)	$f_O = 0$ to 5 kHz		$\pm 0.1\%$			$\pm 0.1\%$			$\pm 0.1\%$			$\pm 0.1\%$		% F.S.
	$f_O = 0$ to 50 kHz		$\pm 0.2\%$			$\pm 0.2\%$			$\pm 0.2\%$			$\pm 0.2\%$		% F.S.
	$f_O = 0$ to 500 kHz		$\pm 0.5\%$			$\pm 0.5\%$			$\pm 0.5\%$			$\pm 0.5\%$		% F.S.
Recovery from power down			100			100			100			100		μs
Response time to output enable (OE)			100			100			100			100		ns

- NOTES: 4. Optical measurements are made using small-angle incident radiation from a light-emitting diode (LED) optical source.
5. The 470 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 470\text{ nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 35\text{ nm}$, and luminous efficacy = 75 lm/W.
6. The 524 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 524\text{ nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 47\text{ nm}$, and luminous efficacy = 520 lm/W.
7. The 565 nm input irradiance is supplied by a GaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 565\text{ nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 28\text{ nm}$, and luminous efficacy = 595 lm/W.
8. The 640 nm input irradiance is supplied by a AlInGaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 640\text{ nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 17\text{ nm}$, and luminous efficacy = 155 lm/W.
9. Irradiance responsivity R_e is characterized over the range from zero to 5 kHz.
10. Saturation irradiance = (full-scale frequency)/(irradiance responsivity).
11. Illuminance responsivity R_v is calculated from the irradiance responsivity by using the LED luminous efficacy values stated in notes 4, 5, and 6 and using $1\text{ lx} = 1\text{ lm}/\text{m}^2$.
12. Nonlinearity is defined as the deviation of f_O from a straight line between zero and full scale, expressed as a percent of full scale.

Copyright © 2007, TAOS Inc.



The LUMINOLOGY® Company

TYPICAL CHARACTERISTICS

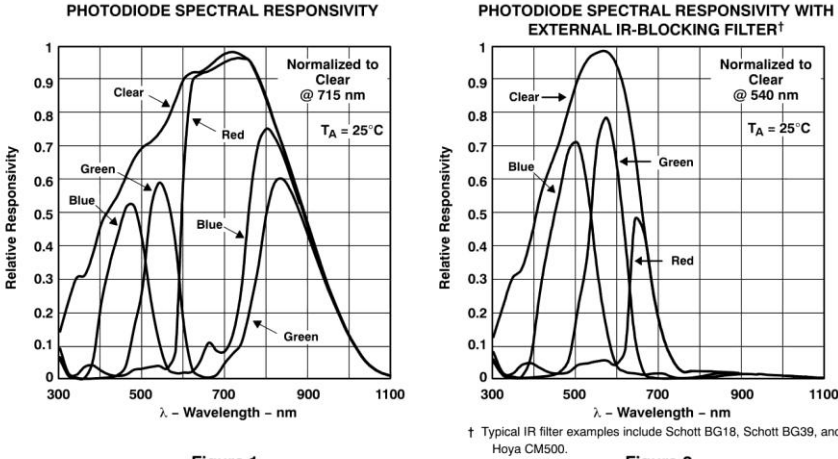


Figure 1

Figure 2

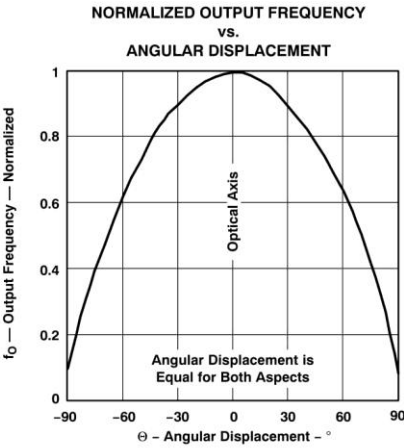


Figure 3

TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
TAOS046M – OCTOBER 2007

APPLICATION INFORMATION

Power supply considerations

Power-supply lines must be decoupled by a 0.01- μ F to 0.1- μ F capacitor with short leads mounted close to the device package.

Input interface

A low-impedance electrical connection between the device $\overline{\text{OE}}$ pin and the device GND pin is required for improved noise immunity.

Output interface

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances. If lines greater than 12 inches are used on the output, a buffer or line driver is recommended.

A high state on Output Enable (OE) places the output in a high-impedance state for multiple-unit sharing of a microcontroller input line.

Powering down the sensor using S0/S1 (L/L) will cause the output to be held in a low state. Because the output is held low, the sensor cannot be powered down in a multiple-unit configuration with a common OUTPUT pin.

Photodiode type (color) selection

The type of photodiode (blue, green, red, or clear) used by the device is controlled by two logic inputs, S2 and S3 (see Table 1).

Output frequency scaling

Output-frequency scaling is controlled by two logic inputs, S0 and S1. The internal light-to-frequency converter generates a fixed-pulsewidth pulse train. Scaling is accomplished by internally connecting the pulse-train output of the converter to a series of frequency dividers. Divided outputs are 50%-duty cycle square waves with relative frequency values of 100%, 20%, and 2%. Because division of the output frequency is accomplished by counting pulses of the principal internal frequency, the final-output period represents an average of the multiple periods of the principle frequency.

The output-scaling counter registers are cleared upon the next pulse of the principal frequency after any transition of the S0, S1, S2, S3, and $\overline{\text{OE}}$ lines. The output goes high upon the next subsequent pulse of the principal frequency, beginning a new valid period. This minimizes the time delay between a change on the input lines and the resulting new output period. The response time to an input programming change or to an irradiance step change is one period of new frequency plus 1 μ s. The scaled output changes both the full-scale frequency and the dark frequency by the selected scale factor.

The frequency-scaling function allows the output range to be optimized for a variety of measurement techniques. The scaled-down outputs may be used where only a slower frequency counter is available, such as low-cost microcontroller, or where period measurement techniques are used.

APPLICATION INFORMATION

Measuring the frequency

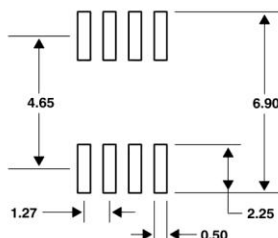
The choice of interface and measurement technique depends on the desired resolution and data acquisition rate. For maximum data-acquisition rate, period-measurement techniques are used.

Output data can be collected at a rate of twice the output frequency or one data point every microsecond for full-scale output. Period measurement requires the use of a fast reference clock with available resolution directly related to reference clock rate. Output scaling can be used to increase the resolution for a given clock rate or to maximize resolution as the light input changes. Period measurement is used to measure rapidly varying light levels or to make a very fast measurement of a constant light source.

Maximum resolution and accuracy may be obtained using frequency-measurement, pulse-accumulation, or integration techniques. Frequency measurements provide the added benefit of averaging out random- or high-frequency variations (jitter) resulting from noise in the light signal. Resolution is limited mainly by available counter registers and allowable measurement time. Frequency measurement is well suited for slowly varying or constant light levels and for reading average light levels over short periods of time. Integration (the accumulation of pulses over a very long period of time) can be used to measure exposure, the amount of light present in an area over a given time period.

PCB Pad Layout

Suggested PCB pad layout guidelines for the D package are shown in Figure 4.



NOTES: A. All linear dimensions are in millimeters.
 B. This drawing is subject to change without notice.

Figure 4. Suggested D Package PCB Layout



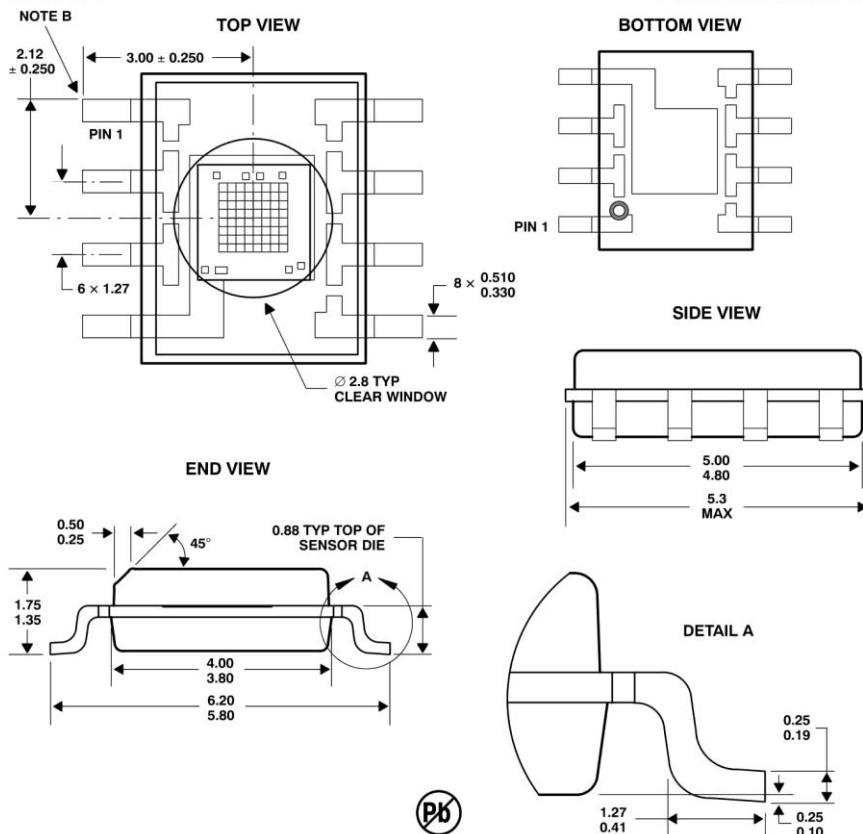
TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
 TAOS046M – OCTOBER 2007

MECHANICAL INFORMATION

This SOIC package consists of an integrated circuit mounted on a lead frame and encapsulated with an electrically nonconductive clear plastic compound. The TCS230 has an 8×8 array of photodiodes with a total size of 1.15 mm by 1.15 mm. The photodiodes are $120 \mu\text{m} \times 120 \mu\text{m}$ in size and are positioned on $144 \mu\text{m}$ centers.

PACKAGE D

PLASTIC SMALL-OUTLINE



- NOTES: A. All linear dimensions are in millimeters.
 B. The center of the 1.15-mm by 1.15-mm photo-active area is referenced to the upper left corner tip of the lead frame (Pin 1).
 C. Package is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
 D. This drawing is subject to change without notice.

Figure 5. Package D — Plastic Small Outline IC Packaging Configuration

Copyright © 2007, TAOS Inc.

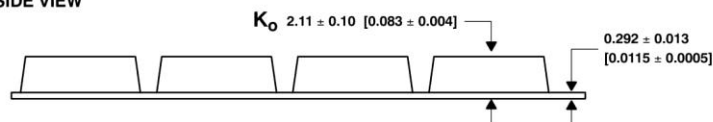


The **LUMINOLOGY**® Company

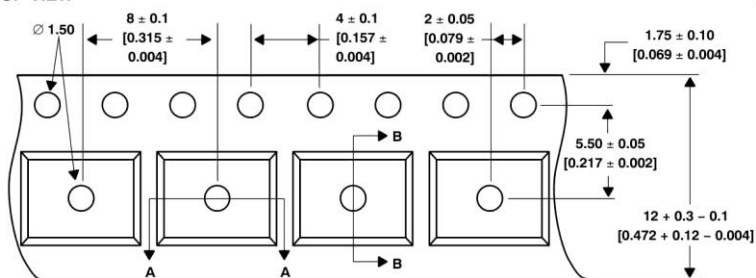
TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
 TAOS046M – OCTOBER 2007

MECHANICAL INFORMATION

SIDE VIEW



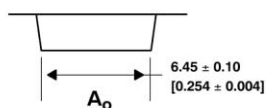
TOP VIEW



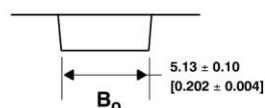
END VIEW



DETAIL A



DETAIL B



- NOTES: A. All linear dimensions are in millimeters [inches].
 B. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
 C. Symbols on drawing A_0 , B_0 , and K_0 are defined in ANSI EIA Standard 481-B 2001.
 D. Each reel is 178 millimeters in diameter and contains 1000 parts.
 E. TAOS packaging tape and reel conform to the requirements of EIA Standard 481-B.
 F. This drawing is subject to change without notice.

Figure 6. Package D Carrier Tape



TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
TAOS046M – OCTOBER 2007

MANUFACTURING INFORMATION

The Plastic Small Outline IC package (D) has been tested and has demonstrated an ability to be reflow soldered to a PCB substrate.

The solder reflow profile describes the expected maximum heat exposure of components during the solder reflow process of product on a PCB. Temperature is measured on top of component. The component should be limited to a maximum of three passes through this solder reflow profile.

Table 2. TCS230 Solder Reflow Profile

PARAMETER	REFERENCE	TCS230
Average temperature gradient in preheating		2.5°C/sec
Soak time	t_{soak}	2 to 3 minutes
Time above 217°C	t_1	Max 60 sec
Time above 230°C	t_2	Max 50 sec
Time above $T_{\text{peak}} - 10^\circ\text{C}$	t_3	Max 10 sec
Peak temperature in reflow	T_{peak}	260°C (-0°C/+5°C)
Temperature gradient in cooling		Max -5°C/sec

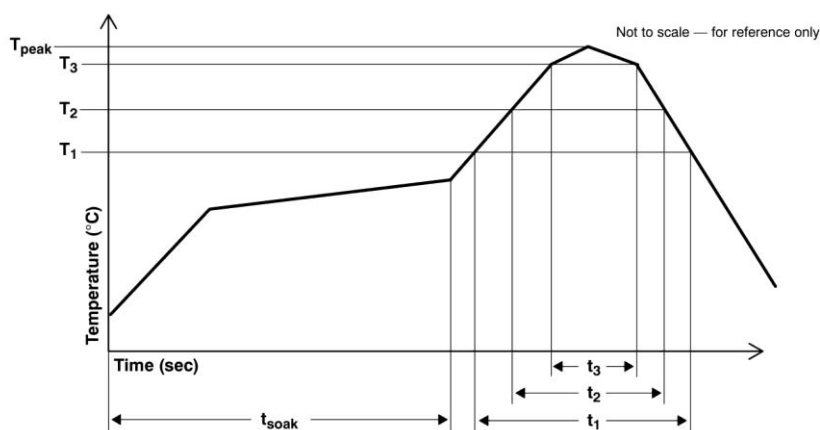


Figure 7. TCS230 Solder Reflow Profile Graph



TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
TAOS046M - OCTOBER 2007

Moisture Sensitivity

Optical characteristics of the device can be adversely affected during the soldering process by the release and vaporization of moisture that has been previously absorbed into the package molding compound. To prevent these adverse conditions, all devices shipped in carrier tape have been pre-baked and shipped in a sealed moisture-barrier bag. No further action is necessary if these devices are processed through solder reflow within 24 hours of the seal being broken on the moisture-barrier bag.

However, for all devices shipped in tubes or if the seal on the moisture barrier bag has been broken for 24 hours or longer, it is recommended that the following procedures be used to ensure the package molding compound contains the smallest amount of absorbed moisture possible.

For devices shipped in tubes:

1. Remove devices from tubes
2. Bake devices for 4 hours, at 90°C
3. After cooling, load devices back into tubes
4. Perform solder reflow within 24 hours after bake

Bake only a quantity of devices that can be processed through solder reflow in 24 hours. Devices can be re-baked for 4 hours, at 90°C for a cumulative total of 12 hours (3 bakes for 4 hours at 90°C).

For devices shipped in carrier tape:

1. Bake devices for 4 hours, at 90°C in the tape
2. Perform solder reflow within 24 hours after bake

Bake only a quantity of devices that can be processed through solder reflow in 24 hours. Devices can be re-baked for 4 hours in tape, at 90°C for a cumulative total of 12 hours (3 bakes for 4 hours at 90°C).



TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
 TAOS046M – OCTOBER 2007

PRODUCTION DATA — information in this document is current at publication date. Products conform to specifications in accordance with the terms of Texas Advanced Optoelectronic Solutions, Inc. standard warranty. Production processing does not necessarily include testing of all parameters.

LEAD-FREE (Pb-FREE) and GREEN STATEMENT

Pb-Free (RoHS) TAOS' terms *Lead-Free* or *Pb-Free* mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TAOS Pb-Free products are suitable for use in specified lead-free processes.

Green (RoHS & no Sb/Br) TAOS defines *Green* to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information and Disclaimer The information provided in this statement represents TAOS' knowledge and belief as of the date that it is provided. TAOS bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TAOS has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TAOS and TAOS suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

NOTICE

Texas Advanced Optoelectronic Solutions, Inc. (TAOS) reserves the right to make changes to the products contained in this document to improve performance or for any other purpose, or to discontinue them without notice. Customers are advised to contact TAOS to obtain the latest product information before placing orders or designing TAOS products into systems.

TAOS assumes no responsibility for the use of any products or circuits described in this document or customer product design, conveys no license, either expressed or implied, under any patent or other right, and makes no representation that the circuits are free of patent infringement. TAOS further makes no claim as to the suitability of its products for any particular purpose, nor does TAOS assume any liability arising out of the use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

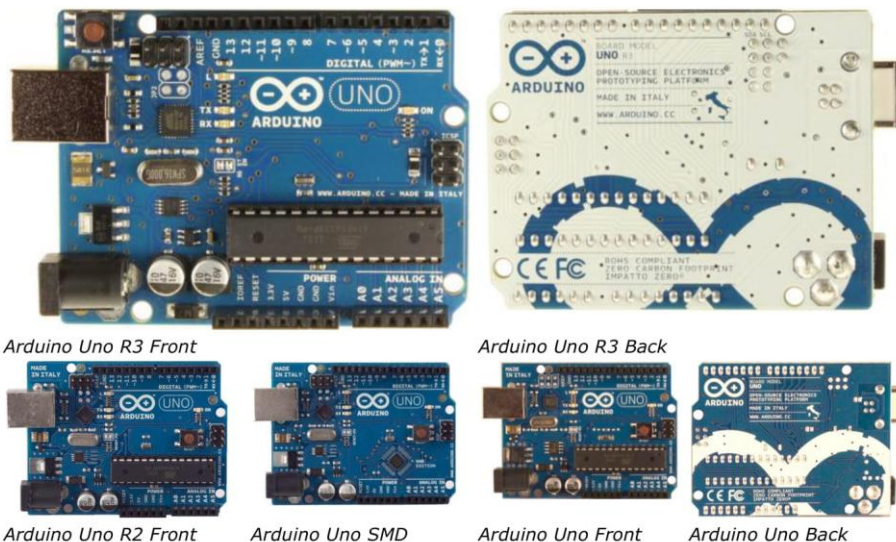
TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS, INC. PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS IN WHICH THE FAILURE OR MALFUNCTION OF THE TAOS PRODUCT MAY RESULT IN PERSONAL INJURY OR DEATH. USE OF TAOS PRODUCTS IN LIFE SUPPORT SYSTEMS IS EXPRESSLY UNAUTHORIZED AND ANY SUCH USE BY A CUSTOMER IS COMPLETELY AT THE CUSTOMER'S RISK.

LUMINOLOGY, TAOS, the TAOS logo, and Texas Advanced Optoelectronic Solutions are registered trademarks of Texas Advanced Optoelectronic Solutions Incorporated.



Lampiran 10. Data Sheet Arduino UNO

Arduino Uno



Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

[Revision 2](#) of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

[Revision 3](#) of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an ATmega8, 168, or 328. Current models use an ATmega328, but an ATmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Lampiran 11. Data Sheet Arduino Ethernet Shield



W5100 Ethernet Shield

-A high performance Ethernet shield for Arduino

Overview



W5100 Ethernet shield is a WIZnet W5100 breakout board with POE and Micro-SD designed for Arduino platform. 5V/3.3V compatible operation voltage level makes it compatible with Arduino boards, leafmaple, and other Arduino compatible board.

Features

- With Micro SD interface
- 5V/3.3V double operational voltage level
- 10Mb/100Mb Ethernet socket with POE
- All electronic brick interface are broken out
- Operation temperature: -40°C ~ +85°C

Specifications

PCB size	55.88mm X 68.58mm X 1.6mm
Indicators	TX,RX,COL,FEX,SPD,LNK
Power supply	5V
Communication Protocol	SPI
RoHS	Yes

Electrical Characteristics

Specification	Min	Type	Max	Unit
Power Voltage	3V	-	5.5	VDC
Input Voltage VH:	3	-	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current	-	-	100	mA

Hardware

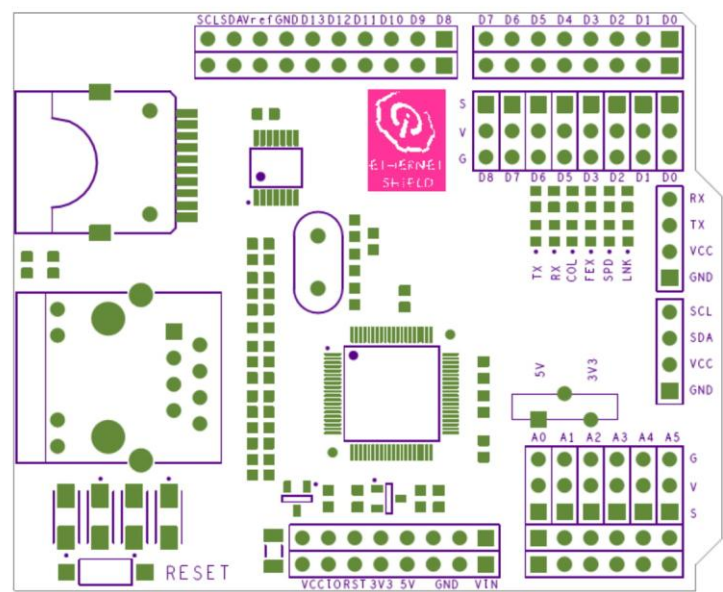


Figure 1 Top Map

Arduino PIN	Description
D0	Rx/Breakout
D1	TX/Breakout
D2	Breakout
D3	Breakout
D4	SD_CS
D5	Breakout
D6	Breakout
D7	Breakout
D8	Breakout
D9	W5100_Reset
D10	W5100_CS
D11	MOSI
D12	MISO
D13	SCK

A0	Breakout
A1	Breakout
A2	Breakout
A3	Breakout
A4	IIC_SDA/Breakout
A5	IIC_SCL/Breakout

Installation

When install W5100 Ethernet shield to Iteaduino, please check the operation voltage level of development board. If the voltage is 3.3V (IFLAT32, Leafmaple), set the Operation Level Setting switch to 3.3V. If the voltage is 5V (Arduino), set the Operation Level Setting switch to 5V.

Iteaduino communicates with both the W5100 and SD card using the SPI bus. This is on digital pins 11, 12, and 13 on the UNO/Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

Indicator LED

The shield contains a number of informational LEDs:

- LNK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FEX: indicates that the network connection is full duplex

- SPD: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COL: flashes when network collisions are detected

Revision History

Rev.	Description	Release date
v1.0	Initial version	2012-09-14

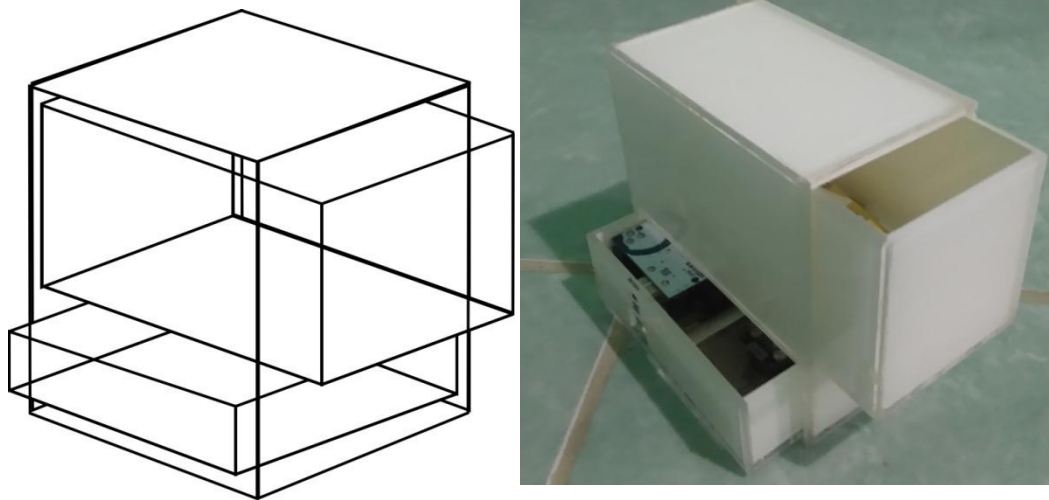
Lampiran 12. Data Pembacaan Sensor

Tabel Data Kalibrasi Sampel Warna Urine

kuning			hijau			merah			coklat			coklat tua			putih		
R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
208	242	237	203	238	225	210	221	231	201	221	225	179	200	207	205	242	255
210	245	243	205	242	231	213	224	237	203	224	231	182	204	213	208	245	261
215	252	255	208	245	237	215	228	243	205	228	237	184	207	219	210	249	279
217	255	261	217	258	273	217	242	267	208	238	261	186	211	225	217	258	285
220	258	267	220	261	279	227	249	273	210	242	267	189	214	255	220	261	291
222	261	273	222	242	231	232	265	303	213	245	237	198	228	261	225	265	297
229	279	279	208	245	237	234	268	309	215	228	261	203	231	279	210	245	321
236	245	303	217	258	273	236	272	315	220	238	267	208	235	303	217	249	261
222	252	255	220	261	279	217	242	267	208	242	225	210	249	219	220	258	279
229	255	261	203	238	225	227	249	273	210	221	231	225	268	225	205	261	285

Nb: nilai pada kotak putih merupakan hasil pembacaan sensor yang paling setabil diantara hasil pembacaan uji coba yang lain, sehingga dijadikan pengkalibrasi sampel pembacaan urine

Lampiran 13. Desain Alat







Gambar Jaring-jaring kerangka *box* alat dan bentuk jadi *box*



Gambar Desain *cutting acrylic box*

Lampiran 14. Foto Alat

	<p>Pengoprasian alat <i>prototype urine analyzer telemetry</i> dengan jaringan <i>localhost</i></p>
	<p>Menghubungkan alat dengan Switch RB250GS dan Router RB450G</p>
	<p>Tamplan <i>wbpage</i> pada <i>client</i></p>
	<p>Menguji sampel urine</p>